

**REC:**  
**Sistema Automático**  
**de Dirección Cinematográfica**  
**en Entornos Virtuales**  
**Basado en Emociones**

---

por Jorge Carrillo de Albornoz  
Trabajo de investigación para el  
Máster en Investigación en Informática  
Especialidad en Sistemas Inteligentes  
Universidad Complutense de Madrid

2008

Dirigido por Pablo Gervás Gómez-Navarro



---

# Agradecimientos

Ante todo me gustaría agradecer el apoyo, ayuda y guía del profesor Dr. Don Pablo Gervás Gómez-Navarro, del Departamento de Ingeniería del Software e Inteligencia Artificial de la UCM (DISIA), sin el cual este trabajo no habría podido ser realidad y del que es director.

Así mismo, me gustaría agradecer a todos los componente del grupo de investigación NIL su ayuda y guía que me han prestado a lo largo de este año, especialmente a Alberto Díaz Esteban, y como no a mi compañero de beca Carlos León Aznar.

Por supuesto, me gustaría agradecer a todo el Departamento de Ingeniería del Software e Inteligencia Artificial su ayuda aportada.

También me gustaría agradecer su apoyo y ayuda incondicional a Laura, que me ha animando en todo momento, y cuya ayuda ha sido indispensable para la realización de esta memoria.

No podría terminar una sección así sin antes agradecer a mi familia su apoyo y esa capacidad para ayudarme en los momento difíciles. También recordar el apoyo de mis amigos que en todo momento han estado a mi lado.





---

# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1. Estructura de la memoria del trabajo de investigación . . . . .	11
<b>2. Introducción a las técnicas cinematográficas</b>	<b>13</b>
2.1. El lenguaje audiovisual . . . . .	13
2.2. Estructura de una película y el guión . . . . .	14
2.3. Tipos de planos . . . . .	16
2.4. Ángulo . . . . .	19
2.5. Línea de interés . . . . .	20
2.6. Posicionamiento de cámara . . . . .	21
2.7. Continuidad . . . . .	22
2.7.1. Temporal . . . . .	22
2.7.2. Espacial . . . . .	23
2.8. Movimientos . . . . .	24
<b>3. Trabajo previo</b>	<b>29</b>
3.1. Los entornos virtuales . . . . .	29
3.1.1. Breve historia de los entornos virtuales . . . . .	30
3.1.2. Aplicaciones de los entornos virtuales . . . . .	31
3.2. Sistemas automáticos de gestión de cámara . . . . .	33
3.2.1. Aplicaciones y gestores de cámara . . . . .	33
3.2.2. Aplicación de técnicas cinematográficas a la gestión de cámara . . . . .	40
3.3. Sistemas de reconocimiento de emociones . . . . .	46
3.3.1. La emoción . . . . .	47
3.3.2. Técnicas en los sistemas de reconocimiento de emociones	49
3.3.3. El sistema Emotag . . . . .	51
<b>4. REC: Sistema Automático de Gestión de Cámara en Entor- nos Virtuales Basado en Emociones</b>	<b>55</b>
4.1. Módulo lógico . . . . .	58
4.1.1. Analizador de Guiones . . . . .	59
4.1.2. Director . . . . .	61

4.2. Módulo gráfico . . . . .	63
4.2.1. Sistema de Visualización de Historias en 3D . . . . .	65
4.2.2. Sistema de Eventos . . . . .	70
4.3. Algoritmo de gestión de cámaras basado en emociones . . . . .	72
4.3.1. Fase I: Análisis del guión . . . . .	75
4.3.2. Fase II: Cálculo de la intensidad emocional . . . . .	77
4.3.3. Fase III: Creación de la historia virtual . . . . .	81
4.3.4. Fase IV: Elección de la configuración de cámara . . . . .	83
4.3.5. Fase V: Representación gráfica de la configuración de cámara . . . . .	87
<b>5. Resultados y evaluación</b>	<b>91</b>
5.1. Simulación de un fragmento de una película . . . . .	91
5.1.1. Secuencias similares y correctas . . . . .	94
5.1.2. Secuencias distintas y correctas . . . . .	96
5.1.3. Secuencias distintas y erróneas . . . . .	98
5.2. Discusión . . . . .	100
5.2.1. Principales aportaciones . . . . .	100
5.2.2. Principales inconvenientes . . . . .	101
<b>6. Conclusiones y trabajo futuro</b>	<b>105</b>
6.1. Estado actual del trabajo . . . . .	105
6.2. Trabajo futuro . . . . .	106
6.2.1. Incorporación del aspecto geométrico . . . . .	107
6.2.2. Análisis de los diversos métodos para el cálculo de la intensidad . . . . .	107
6.2.3. Estudio de la aplicación de grafos a los sistemas de emociones . . . . .	108
6.2.4. Generación de transiciones suaves . . . . .	108
6.2.5. Aumento del tipo de secuencias . . . . .	109
6.2.6. Adaptación del algoritmo para entornos interactivos . . . . .	109
6.2.7. Mejora del sistema de carga del módulo gráfico . . . . .	110
<b>A. Ejemplos de reglas en el sistema</b>	<b>121</b>
<b>B. Fragmento del guión de la película</b>	<b>131</b>
<b>C. Publicaciones</b>	<b>133</b>

---

# Índice de figuras

2.1. Esquema del formato estándar de los guiones. . . . .	15
2.2. Clasificación de planos largos usados en el sistema . . . . .	17
2.3. Clasificación de planos medios usados en el sistema . . . . .	18
2.4. Clasificación de planos cortos usados en el sistema . . . . .	18
2.5. Clasificación de ángulos . . . . .	20
2.6. Posicionamiento de cámara. . . . .	21
2.7. Regla de los 180° . . . . .	24
2.8. Panorámica Horizontal o <i>Pan</i> . . . . .	25
2.9. Panorámica Vertical o <i>Tilt</i> . . . . .	26
2.10. <i>Zoom</i> . . . . .	26
2.11. <i>Dolly</i> . . . . .	27
3.1. Lenguaje DCCL correspondiente a una conversación entre dos personas. . . . .	42
3.2. Mapa bidimensional de espacio emocional de Watson y Tellegen. . . . .	48
3.3. Clasificación de Sistemas de Anotación Semántica según sus métodos. . . . .	50
4.1. Arquitectura del sistema . . . . .	57
4.2. Elementos de <i>Analizador de Guiones</i> . . . . .	60
4.3. Elementos del <i>Director</i> . . . . .	61
4.4. Ejemplo de la sintaxis de reglas del sistema Drools para este sistema. . . . .	62
4.5. Esquema Módulo Gráfico . . . . .	64
4.6. Jerarquía de clases de los diferentes objetos de la aplicación . . . . .	67
4.7. Jerarquía de clases de las diferentes acciones de la aplicación . . . . .	69
4.8. Formato HMLT exportado por la aplicación Celtx. . . . .	75
4.9. Estructura de datos del guión en formato XML. . . . .	77
4.10. Emociones básicas manejadas por <b>Emotag</b> . . . . .	79
4.11. Estructura de datos del guión enriquecida con la intensidad emocional. . . . .	81
4.12. Estructura de datos de la historia en formato XML. . . . .	82
4.13. Eventos del sistema. . . . .	84

4.14. Configuraciones de cámara seleccionadas por el sistema. . . .	86
5.1. Aragorn y Eowyn . . . . .	92
5.2. Gimli y Pippin . . . . .	93
5.3. Secuencias de la simulación similares y correctas . . . . .	95
5.4. Secuencias de la película similares y correctas . . . . .	95
5.5. Secuencias de la simulación diferentes y correctas . . . . .	97
5.6. Secuencias de la película diferentes y correctas . . . . .	97
5.7. Secuencias de la simulación diferentes y erróneas . . . . .	99
5.8. Secuencias de la película diferentes y erróneas . . . . .	99
A.1. Ejemplo 1 de las reglas de las secuencias de acción del sistema.	121
A.2. Ejemplo 2 de las reglas de las secuencias de acción del sistema.	122
A.3. Ejemplo 3 de las reglas de las secuencias de acción del sistema.	123
A.4. Ejemplo 4 de las reglas de las secuencias de acción del sistema.	124
A.5. Ejemplo 5 de las reglas de las secuencias de acción del sistema.	125
A.6. Ejemplo 6 de las reglas de las secuencias de acción del sistema.	126
A.7. Ejemplo 1 de las reglas de las secuencias de diálogo del sistema.	127
A.8. Ejemplo 2 de las reglas de las secuencias de diálogo del sistema.	128
A.9. Ejemplo 3 de las reglas de las secuencias de diálogo del sistema.	129
A.10. Ejemplo 4 de las reglas de las secuencias de diálogo del sistema.	130

---

# Capítulo 1

## Introducción

Una de las principales funciones de la fotografía y de la cinematografía consiste en obtener la manera óptima de representar y transmitir la información mediante imágenes. El posicionamiento de cámaras, desde el punto de vista del director de fotografía, y el movimiento de las mismas, desde el punto de vista del director, son dos actividades orientadas a determinar la información que se debe mostrar y el modo de conseguirlo. En el mundo de la computación gráfica se han propuesto diversas técnicas y aproximaciones para lograr estas funcionalidades en entornos virtuales, mediante el posicionamiento y movimiento automático de las cámaras virtuales. Los sistemas de control automático de cámaras cubren un amplio abanico de tareas en multitud de problemas de computación gráfica y aplicaciones, incluyendo la visualización de información, inspección de objetos distantes o próximos, narratología o juegos en 3D. Sin embargo, a diferencia de otros campos, los sistemas de control de cámaras han recibido poca atención por parte de la comunidad científica.

Sorprendentemente, aunque las técnicas cinematográficas proponen un conjunto amplio de nociones que describen diversas aproximaciones de posicionamiento, orientación y movimiento de las cámaras, para conseguir una mejor expresividad e impacto sobre el espectador, es poco frecuente ver este tipo de conceptos en los diferentes campos de la informática gráfica.

Es por esto que en los últimos años se han realizado diversos trabajos que intentan adaptar y representar el conocimiento cinematográfico sobre la gestión de cámaras en los diversos tipos de entornos virtuales. Aunque es cierto que este campo todavía se encuentra en una fase previa de investigación, por lo que su aplicación no se ha llevado a cabo aún en las diversas aplicaciones comerciales en las que tendría cabida, como los videojuegos, sistemas educativos, etc., actualmente este tipo de nociones empiezan a cobrar un alto interés en las aplicaciones, en las que se requiere cada vez más una experiencia casi real.

Sin embargo, la mayoría de estas aproximaciones se centran más en el

aspecto geométrico de la escena que en el aspecto dramático y emotivo característico de las técnicas cinematográficas, intentando evitar así, mediante la aplicación de los diferentes conceptos cinematográficos, situaciones en las que no se obtiene una correcta visualización de los objetos importantes de la escena, como por ejemplo conversaciones de varios personajes en las que la disposición geométrica de éstos requiere de diferentes planos para una correcta interpretación, o tomas en las que un objeto se interpone entre la cámara y el personaje, u objeto, principal.

Los pocos trabajos que han introducido en cierta forma un aspecto más dramático de las diferentes técnicas cinematográficas no resuelven el problema de cómo obtener la información necesaria para la gestión adecuada de las cámaras, aplicando en todos los casos configuraciones preestablecidas sobre las acciones de los personajes del entorno virtual, como por ejemplo, felicidad a la acción de reírse, o mediante estados de ánimo asociados a los personajes que varían según sus acciones, o generando a mano los diferentes parámetros previo estudio de los acontecimientos de la historia.

Además, algo que no se suele precisar en este tipo de trabajos, o que se suele realizar al igual que con la carga emotiva mediante configuraciones asociadas a acciones, es la obtención de la información necesaria para determinar qué es lo que está sucediendo en la escena, como una conversación o una acción de un personaje.

Atendiendo a esto, parece interesante el estudio de un sistema automático de dirección cinematográfica basado en emociones, capaz de determinar en tiempo real qué es lo que está sucediendo en la escena, así como la forma emotiva en la que sucede, y seleccionar la mejor forma de visualizarlo mediante la aplicación de las diferentes técnicas utilizadas en el mundo del cine.

Es por esto que en este trabajo se propone la utilización de diversos métodos de Procesamiento del Lenguaje Natural sobre el guión de la historia, para obtener la información necesaria en la gestión automática de las cámaras del entorno virtual, interpretando a su vez las diferentes nociones cinematográficas desde un punto de vista más dramático y emotivo, y pretendiendo obtener de esta forma un mayor impacto sobre el espectador.

Para poder realizar esto, el sistema debe ser capaz de identificar los diferentes elementos de los que se compone un guión, así como determinar las emociones inmersas en los textos de dichos elementos, generando con esta información una estructura de la historia con los diferentes atributos necesarios para la correcta elección de la mejor configuración de cámara. A partir de esta estructura, el sistema podrá determinar en tiempo real, junto con un conjunto de eventos que permitan la sincronización con el entorno virtual, la mejor configuración de cámara para lo que está sucediendo en la historia en cada momento.

De esta manera, el sistema será capaz de convertir la información conceptual contenida en los textos de los guiones en las operaciones geométricas

necesarias para la gestión de cámaras en el mundo virtual de los motores gráficos, pasando así de un plano abstracto, como es el lenguaje natural del guión, a un plano concreto, como son las diferentes posiciones y movimientos de las cámaras, y consiguiendo toda la expresividad inmersa en las técnicas de cine aplicadas.

## 1.1. Estructura de la memoria del trabajo de investigación

La presente memoria sobre el trabajo de investigación que se ha realizado está dividida en seis capítulos, a través de los cuales se pretende dar una visión clara del sistema propuesto, detallando los aspectos más importantes e interesantes.

- En el Capítulo 1, **Introducción**, se introduce el sistema exponiendo los diferentes aspectos que han motivado esta investigación y los objetivos que se pretenden conseguir.
- En el Capítulo 2, **Introducción a las técnicas cinematográficas**, se realiza un estudio de las diferentes técnicas cinematográficas en las que se basará parte del sistema.
- En el Capítulo 3, **Trabajo previo**, se expone una visión de las diferentes aproximaciones previas en las que se ha basado e inspirado este trabajo.
- En el Capítulo 4, **REC: Sistema Automático de Gestión de Cámara en Entornos Virtuales Basado en Emociones**, se describen y detallan los diferentes componentes del sistema desarrollado, a la vez que se expone el algoritmo de gestión de cámaras basado en emociones.
- En el Capítulo 5, **Resultados y evaluación**, se muestran y discuten los resultados obtenidos con el sistema en una simulación de un fragmento de una película, concluyendo con un análisis de las diferentes aportaciones e inconvenientes encontrados.
- El Capítulo 6, **Conclusiones y trabajo futuro**, se presentan las conclusiones extraídas de la realización del trabajo y los resultados obtenidos, proponiendo diferentes trabajos futuros para la mejora del sistema.
- Se completa el trabajo con la Bibliografía y los Apéndices A, **Ejemplos de reglas en el sistema**, B, **Fragmento del guión de una película**, y C, **Publicaciones**.





---

## Capítulo 2

# Introducción a las técnicas cinematográficas

La cinematografía se define como el arte de hacer películas. En este campo se encapsulan las técnicas y principios que gobiernan cómo debe filmarse una acción o acontecimiento mediante imágenes. Más en particular, proporciona una guía sobre cómo se debe usar la cámara para conseguir potenciar el interés del espectador, y obtener una mayor expresividad y claridad del texto narrativo que se desea contar, presentándolo de una manera interesante y coherente.

A lo largo del capítulo se va a presentar una introducción a las técnicas cinematográficas más relevantes, así como a los conceptos que implican, y que han servido de inspiración para la realización de parte de este trabajo. Se comenzará describiendo qué es lo que se entiende por lenguaje audiovisual, y se expondrá la visión de los autores más destacados, para después mostrar la manera en que se estructuraran películas y el formato específico de los guiones. Finalmente, se presentarán los diferentes parámetros que entran en juego en este lenguaje, y que permiten dotar de esa expresividad al mundo del cine.

### 2.1. El lenguaje audiovisual

Si algo se puede decir del cine, desde el punto de vista técnico, dramático y narrativo, es la gran diversidad y riqueza que a lo largo de los años ha conseguido para expresar, desde muchas perspectivas diferentes, un mismo acontecimiento, cada una con sus matices y características representativas. Sin embargo, la gran mayoría de estas técnicas se pueden clasificar según un conjunto de propiedades, con sus diferentes parámetros cada una, que proporcionan ese amplio espectro del que tan característico es el séptimo arte. A lo largo de los últimos años del siglo XX, un grupo numeroso de autores propusieron diversas visiones sobre este conjunto de técnicas, y las

diferentes aplicaciones de éstas según la secuencia filmada.

Los dos autores más importantes y citados en este campo son Arijon en (1991 (originally published 1976)) y Mascelli en (1965). En el primero, se presenta una clasificación de las distintas situaciones que pueden darse en una escena, así como sus componentes y características principales, y se muestran las distintas alternativas de tomas y restricciones sobre éstas. Para Arijon sólo existen tres tipos simplificados de escenas : acción, diálogo y ambas a la vez, y en cada caso, dos conjuntos de técnicas a utilizar: aquellas que implican movimiento de la cámara y las que no. Mascelli, por el contrario, basa sus distintas configuraciones de cámara en principios narrativos y de continuidad espacial y temporal, agrupando las distintas perspectivas en que puede ser filmada una secuencia.

Este conjunto de configuraciones que describen ambos autores se basa en parámetros como la distancia al objeto principal, los tipos de movimientos, la línea de visión, las diferentes posiciones de la cámara respecto a una situación, el encuadre de la escena o el ángulo de orientación, mediante los que se gestionan las distintas aproximaciones más utilizadas, y que representan una mayor expresividad y claridad de la historia que se desea contar. En ciertas ocasiones, estas reglas son quebrantadas a propósito por los directores para conseguir inducir en el espectador una emoción o sentimiento deseado, como por ejemplo *shock*, confusión, etc.

## 2.2. Estructura de una película y el guión

A lo largo de los años, se ha ido estableciendo en el mundo del cine una estructura estándar para representar las diferentes etapas y elementos que componen la creación y el diseño de una película, y que representa una forma coherente y aceptada de presentarla. Por supuesto, no sería correcto decir que todas las producciones se limitan a un formato concreto, y más en un campo en el que la palabra arte toma una gran relevancia, ya que esto no habría dotado al cine de la amplia diversidad y riqueza que posee. Sin embargo, por regla general, los diferentes elementos de esta estructura se pueden encontrar, en mayor o menor medida, en casi todos los géneros que se han producido, y aunque en cada caso estos elementos suelen estar matizados y caracterizados según la visión de sus creadores, se puede hablar de un formato que generaliza este proceso.

Así, se puede decir que una película se divide en un conjunto de *secuencias*, donde cada una comprende una unidad narrativa y está compuesta por un conjunto de *escenas*, relacionadas mediante el mismo momento dramático o narrativo. A su vez, cada *escena* está formada por un conjunto de *tomas*, y representa una unidad dramática en la cual la acción transcurre en un mismo lugar o período de tiempo. La *toma* se define como la unidad más pequeña en esta estructura, y representa el intervalo de visión grabado con

una misma cámara y sin interrupción. Esto se puede realizar mediante planos, posiciones estáticas de la cámara, o mediante movimientos, dependiendo del contenido de la escena y de lo que se pretenda expresar.

Para poder representar esta estructura en un formato escrito, se establecieron un conjunto de normas a la hora de escribir el texto de la obra, mediante las que se podían identificar los distintos elementos de ésta, y que enriquecían y ayudaban a los múltiples componentes del equipo de la película. Este conjunto de normas se establecieron como estándar con el paso de los años, facilitando la comprensión de los textos de los guiones. Esto ha propiciado que en los últimos años hayan surgido numerosos programas informáticos que facilitan la creación y gestión de estos textos, automatizando parte del proceso y permitiendo exportar este estándar a formatos más manejables por los ordenadores que el texto plano.

```
INT. DRISKILL HOTEL SEMINAR ROOM DAY

JOE and APRIL burst through the doors into a clean,
well-lit seminar room.

                JOE
            Are we in time?

                APRIL
            How could they start without us?
            We're the main attraction.

Joe catches his breath as he leans against the podium at
the front of the room.

                JOE
            (looking about the room)
            We are?

                APRIL
            Don't be a moron. You know we've
            been invited to Austin to discuss
            script format.
```

Figura 2.1: Esquema del formato estándar de los guiones.

En la Figura 2.1 se puede ver un ejemplo de este estándar, en el que se pueden observar los distintos formatos para representar los elementos que lo conforman. Las diferentes *escenas* en los guiones se representan mediante frases en letra mayúscula y alineadas a la izquierda, normalmente precedidas de expresiones como “EXT” o “INT”, exterior e interior respectivamente, que representan el lugar y momento en el que se desarrolla la acción. Tras esto suele comenzar una *acción*, mediante la que se describe la escena y lo que está sucediendo en ella, como un cambio de posición de los personajes, un evento nuevo, la entrada de un actor, etc., y que cuya posición está alineada a la izquierda. Estos elementos suelen ir intercalados entre los *diálogos*, siendo

muy aclaratorios a la hora de manejar y visualizar la escena. Las acciones y su calidad descriptiva dependen en gran medida del guionista y director. Los *diálogos* están representados en los guiones como elementos alineados en el centro, que se componen de una línea en letra mayúscula que representa al *actor* y el párrafo siguiente, que representa el *diálogo* de éste.

Para dar mayor riqueza y facilidad de comprensión a estos textos, se suelen usar en los diálogos los *paréntesis*, que se pueden interpretar como pequeñas *acciones* que describen el estado emocional del *actor*. Estos últimos se introducen mediante letras mayúsculas en su primera aparición en el guión, pudiéndose escribir en minúsculas en posteriores apariciones en las *acciones*, y siempre en mayúsculas en los *diálogos*.

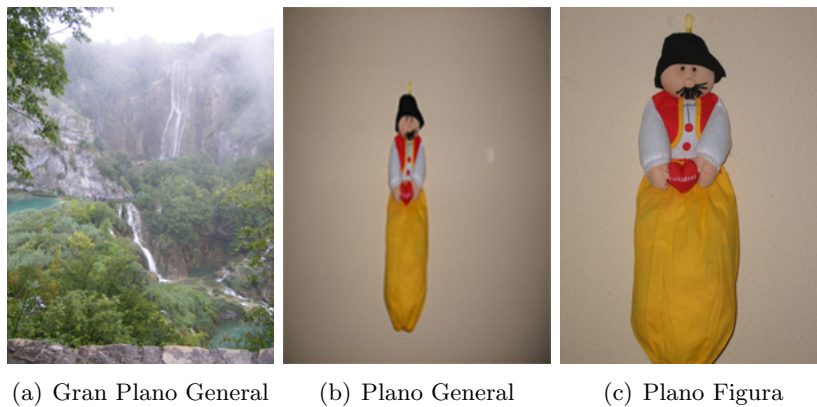
## 2.3. Tipos de planos

El plano es uno de los elementos más representativos de los lenguajes audiovisuales, y uno de los que mayor capacidad expresiva y emotiva tiene. Como dijo Hitchcock, “El tamaño de un objeto en una escena es igual a la importancia de éste en la historia”. Esta importancia se representa mediante la distancia a la que es filmado el objeto, es decir, cuanto más lejos se presenta éste, menos relevante para la escena es. Por el contrario, cuando se quiere enfatizar algo o expresar una alta carga emotiva, se utilizan distancias cortas.

A lo largo de los años, ha habido gran diversidad en la clasificación de los diferentes planos, así como en su nomenclatura. Aún a día de hoy, es difícil encontrar un consenso en esta materia, variando las distintas opciones de un autor a otro. Por ejemplo, Arijon en (1991 (originally published 1976)) propone 5 categorías de planos, mientras que Millerson en (1983) propone 10 y Hawkins 13 en su sistema multiagente (2005). Un aspecto común, a casi todas las aproximaciones, es representar las distintas categorías respecto al tamaño de una figura humana.

En lo que si que no hay duda, aunque hablando de técnicas de cine no hay una regla escrita, es que el aspecto emotivo y dramático de una escena suele presentarse con planos más cortos y cercanos al personaje, y que los más alejados suelen representar mayor carga narrativa y permiten orientar al espectador en la escena. Aunque, como en todo, hay excepciones, y como se verá a continuación, ciertos planos largos son utilizados con carga dramática.

En este trabajo se ha hecho una síntesis de las diversas posibilidades centrándose en las 9 que se describen a continuación. En las Figuras 2.2, 2.3 y 2.4 se pueden ver los distintos planos y su representación visual. Se han diferenciado tres grupos: planos largos, medios y cortos. Los primeros son planos introductorios en la acción, y suelen utilizar para establecer la acción que está sucediendo en la escena. Dentro de los planos largos, a su vez, se pueden distinguir:



(a) Gran Plano General

(b) Plano General

(c) Plano Figura

Figura 2.2: Clasificación de planos largos usados en el sistema

1. **Gran Plano General:** Este plano suele ser utilizado con un carácter descriptivo, representando un amplio paisaje, y mostrando al espectador el lugar en donde se centrará la historia. En ciertas ocasiones, también puede representar soledad o fatalidad. Otro uso muy común se puede encontrar en acciones masivas, como batallas.
2. **Plano General:** Este tipo de planos suele contener un alto valor narrativo, orientando al espectador en la escena, y muy utilizado como plano introductorio en las secuencias. Representa el encuadre de una persona que ocupa un tercio del espacio.
3. **Plano Figura:** El plano figura tiene un valor narrativo, aunque potencia el valor dramático al centrarse en una sola persona. Representa el encuadre de una persona en la imagen.

Los planos medios, por regla general, suelen acompañar a escenas con movimiento de personajes, en los que es necesario situar al actor en la escena, a la vez que se incrementa en cierto grado la carga dramática de la situación y se muestra la acción realizada. Dentro de los planos medios, a su vez, se pueden distinguir:

1. **Plano Medio Largo:** Este plano representa el encuadre de una persona de pantorrilla para arriba en la imagen, potenciando un poco el valor dramático de la escena.
2. **Plano Americano:** Este plano es uno de los más utilizados en las películas del oeste, donde se necesitaba encuadrar a una persona de rodillas para arriba para poder presentar los duelos, mostrando las manos en las pistolas. Tiene un carácter narrativo, presentando a ambos personajes en la acción, pero con cierto dramatismo.



Figura 2.3: Clasificación de planos medios usados en el sistema

3. **Plano Medio:** Representa el encuadre de una persona de cintura para arriba, y suele ser utilizado en diálogos entre personas a una cierta distancia, y donde la carga dramática no es muy alta.

Los planos cercanos representan una enfatización de una acción o evento, es como decir “Mire aquí, esto es importante”, aumentando sensiblemente la intensidad emotiva según se acorta la distancia al actor. Este tipo de planos suelen ser los más utilizados, intercalándolos con planos medios y largos, que reorienten al espectador en la escena. Dentro de los planos cortos, a su vez, se pueden distinguir:

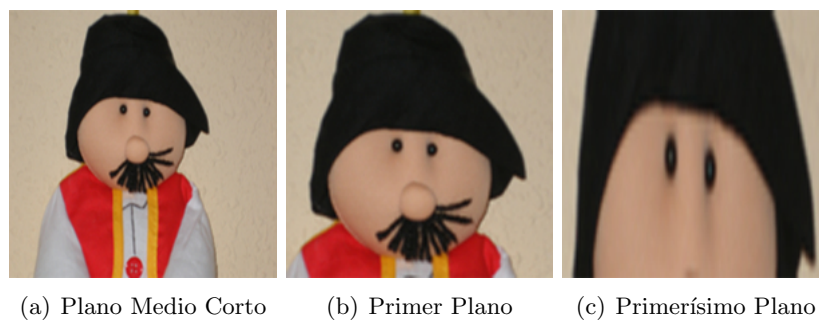


Figura 2.4: Clasificación de planos cortos usados en el sistema

1. **Plano Medio Corto:** Este plano tiene una alta connotación dramática, y representa a una persona de busto para arriba en la imagen.
2. **Primer Plano:** Este plano es el que mayor carga dramática suele conllevar, con un alto valor expresivo y psicológico. Representa el encuadre

de la cara de una persona.

3. **Primerísimo Plano:** Plano muy descriptivo para centrar la atención del espectador en un objeto o elemento concreto. Suele conllevar un alto valor descriptivo, y representa un objeto determinado en la pantalla, como una llave en una mano o los ojos de una persona.

Se ha elegido este conjunto de planos por ser lo suficientemente característico de las distintas aproximaciones, y el cual permite un amplio rango de posibilidades a la hora de interpretar una situación específica dentro de una escena, tanto desde el punto de vista emotivo como desde el geométrico.

Para poder representar las distancias adecuadamente respecto al personaje, de carácter humano o no, un sistema virtual debería tratar cada plano como un número respecto a la distancia. Por regla general los entornos virtuales definen las unidades de sus coordenadas en metros, por lo que se podría tomar esta unidad como representativa de la distancia en los planos. Sin embargo, si algún sistema tomara otra unidad, como el centímetro o milímetro, un simple reajuste en la escala del sistema sería necesario para una correcta interpretación.

## 2.4. Ángulo

El ángulo es otro de los factores que influyen en cómo una secuencia es interpretada por un espectador, y gracias a él se pueden caracterizar a los personajes y su psicología, por ejemplo, ensalzando su aspecto divino con ángulos menores de  $90^\circ$  respecto de la línea formada por el personaje y el suelo. Las distintas posibilidades de este parámetro pueden afectar al ritmo de la escena, a la intensidad dramática, etc. Existen muchas configuraciones posibles, cada una con sus matices característicos. A continuación se van a mostrar las más utilizadas, junto con una descripción y análisis sobre su uso. Un ejemplo visual de esta clasificación se puede ver en la Figura 2.5.

- **Normal:** En esta configuración la cámara se sitúa justo a la altura de los ojos de los personajes. Es una de las configuraciones más comunes para aquellas escenas que no conllevan carga dramática, y que suelen servir como transiciones entre las que tienen gran intensidad emotiva.
- **Picados:** Estos ángulos son los que presentan una amplitud mayor de  $90^\circ$  entre la línea formada por el personaje y el suelo. Este tipo de configuración es muy usada para restar importancia a los elementos o personajes de la escena, e incluso como una perspectiva de un ser divino. Normalmente, se utiliza con ángulos no muy superiores a  $90^\circ$ , para aumentar la carga dramática y mejorar la perspectiva del espectador. Este ángulo con una configuración de  $180^\circ$  se suele denominar “vista de pájaro”.



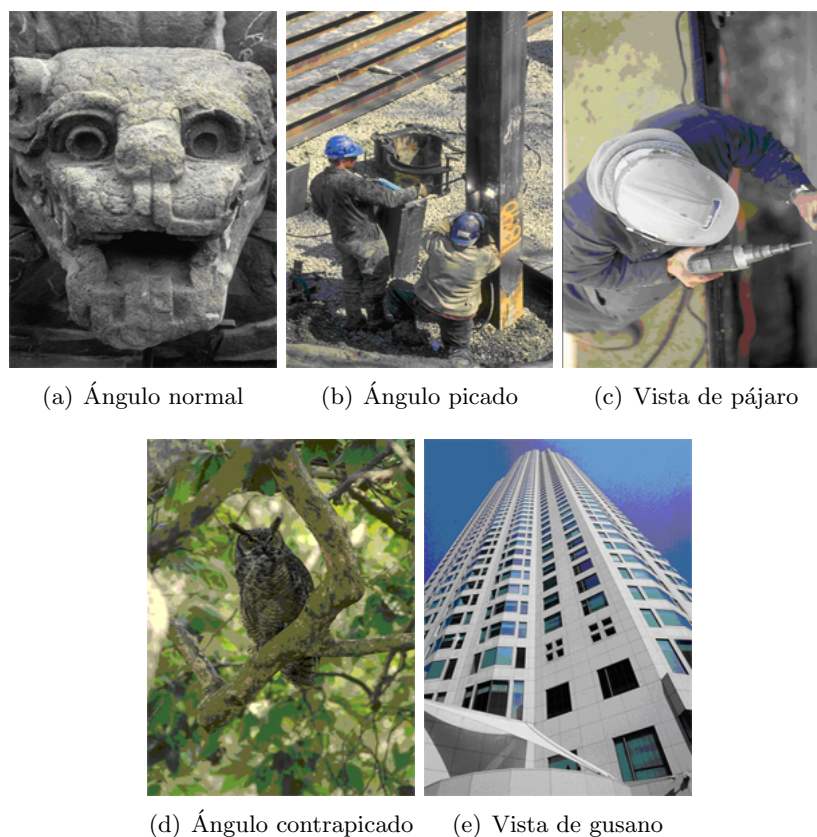


Figura 2.5: Clasificación de ángulos

- **Contrapicado:** Por el contrario, estos ángulos, menores de  $90^\circ$  con la línea del personaje y el suelo, producen una sensación de exaltación de los elementos o personajes de la escena, ensalzando así su superioridad. Esta configuración con un ángulo cercano a los  $0^\circ$  se suele denominar “vista de gusano”.

## 2.5. Línea de interés

La línea de interés o eje de acción es otro de los elementos fundamentales de las técnicas de cine, aunque no es utilizado para todas las secuencias. Para explicar este concepto, se suele hacer uso de las escenas de diálogo entre dos actores, donde la línea de visión queda definida como la línea que une las miradas de ambos. En la Figura 2.6 se muestra este concepto. Existen muchas configuraciones según las distintas posiciones de los actores dependiendo, por ejemplo, de si están situados de espaldas, en paralelo, o a diferente distancia visual (uno sentado y otro de pie). La línea de visión



se define en las acciones de movimiento como la dirección en la que éste se está produciendo.

Este concepto es muy importante porque en ningún caso la línea de visión debe ser atravesada por las cámaras una vez que se ha empezado a filmar desde un lado del eje de acción. Es lo que se conoce como la regla de los  $180^\circ$  y será explicada más adelante en la sección 2.7.2 de continuidad espacial.

## 2.6. Posicionamiento de cámara

Las diferentes técnicas sobre el posicionamiento de cámaras ligan directamente la línea de interés, entre dos o más actores, con las diferentes posiciones de la cámara. Este conjunto de posiciones predefinidas se suelen utilizar para determinadas situaciones de escenas, como por ejemplo, los diálogos, aunque su uso puede ser extendido a cualquier circunstancia en la que tomen parte personajes. De esta manera, se engloban las posibles opciones dentro de 4 grupos principales, que se muestran en la Figura 2.6.

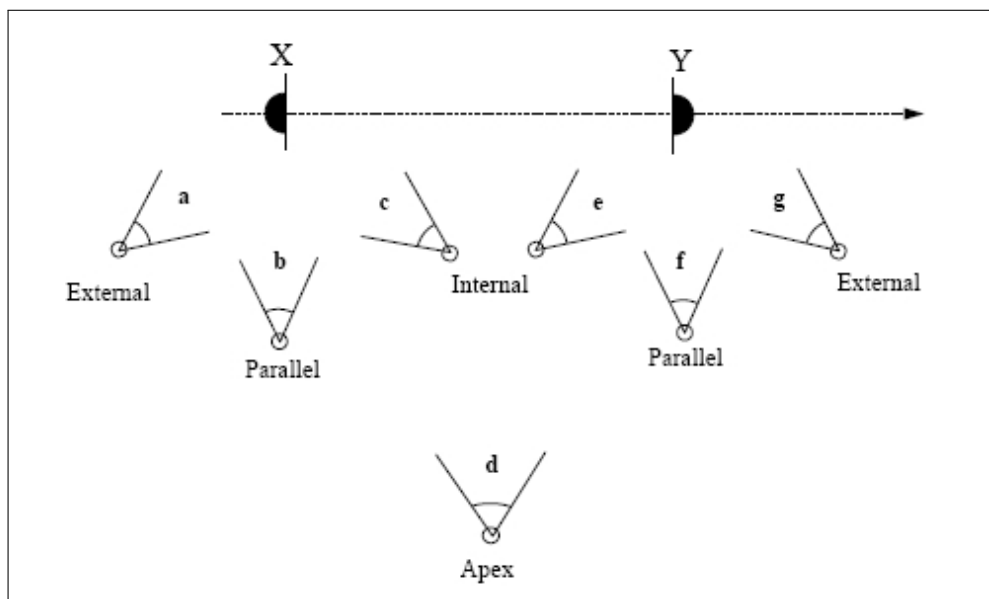


Figura 2.6: Posicionamiento de cámara.

- **Externos:** Las figuras *a* y *g* muestran el posicionamiento de la cámara detrás del personaje, típico en el plano “*over the shoulder*”, formándose un ángulo mayor a  $90^\circ$  con la línea de visión.
- **Paralelos:** En las figuras *b* y *f*, la posición es paralela al actor que se desea enfocar, formando un ángulo de  $90^\circ$  con la línea de visión.

- **Internos:** En las figuras *c* y *e*, la posición es interior a la perpendicular con la línea de visión, formando un ángulo menor de  $90^\circ$  con la línea de visión.
- **Apex:** La figura *d* es la posición en la que se muestran los dos, o más actores, de una conversación.

Este conjunto de posiciones suele denominarse “el principio del triángulo”, y puede extenderse a situaciones con tres o más personas, donde la cámara se centra en el conjunto del grupo desde una posición *d*, o se seleccionan los personajes dos a dos, utilizando el resto de configuraciones posibles. Esto se verá con más detalle en la sección 2.7.2.

En las secuencias en las que solo entra en juego un elemento o un actor se suele mostrar a éste en el centro de la pantalla. Para ello se suele dividir la escena en una cuadrícula de 3x3, situando en las 3 filas centrales al personaje, dependiendo del plano utilizado.

## 2.7. Continuidad

El proceso de creación de una película es muy diferente al resultado final que se puede ver en la pantalla. Normalmente los equipos de rodaje graban las secuencias sin ningún orden, para más tarde unir los distintos fragmentos en el proceso de postproducción. Además, por regla general, no se utiliza una sola cámara para rodar una misma escena, sino que se graba desde muchos puntos de vista diferentes para lograr captar el más adecuado a la historia. Esto implica que todas las escenas deben disponerse en un orden lógico, tanto temporal como espacial, sin producir cortes que harían que el espectador perdiera el hilo de la historia o tuviera que buscar por toda la pantalla al actor que estaba a la izquierda, y que en el siguiente plano está situado a la derecha.

Debido a esto, la continuidad, espacial y temporal, es otro de los factores importantes a la hora de reproducir, mediante una sucesión de imágenes, una escena. Si bien es cierto que este tipo de problemas son casi inexistentes, o muy reducidos, en los entornos virtuales, ya que tanto el tiempo como las acciones son continuas, es decir no se producen cortes en su ejecución, son en muchos casos muy beneficiosas para obtener una mayor calidad de las imágenes mostradas por las cámaras virtuales.

A continuación se van a mostrar las distintas técnicas que se han adoptado en el mundo del cine para evitar este tipo de cortes y confusiones en las secuencias.

### 2.7.1. Temporal

La continuidad temporal quizá sea uno de los parámetros que menos importancia tiene desde el punto de vista de las técnicas de cámara, ya que la

principal carga de esta tarea se realiza en el proceso de edición. Sin embargo, este factor es muy importante a la hora de conseguir que el espectador tenga la sensación de que la historia sigue un hilo temporal correcto, y que no suceden acciones imposibles, como por ejemplo, que un personaje andando entre en la escena por la izquierda, se haga un plano corto sobre él durante unos segundos, y al volver al plano largo el actor esté situado a la izquierda de nuevo, sin haber avanzado nada en el tiempo transcurrido, suponiendo gran confusión y pérdida del espacio temporal en el espectador.

Para evitar esta situación, los equipos de rodaje suelen filmar las distintas escenas desde muchos ángulos distintos, para poder elegir en el proceso de edición el más adecuado. Por el contrario, en los entornos virtuales este tipo de situaciones no pueden darse, ya que no se producen cortes en el proceso de filmación de las escenas.

### 2.7.2. Espacial

Para evitar las situaciones descritas anteriormente, se han adoptado a lo largo de los años un conjunto de técnicas que permiten reducir en gran medida posibles confusiones a la hora de visualizar la película. Para ello, se toma como elemento principal la línea de interés, descrita anteriormente en la sección 2.5, sobre la que se han creado un conjunto de reglas que mejoran la calidad y evitan los cortes anómalos.

#### Regla de los 180°

En esta regla se divide la escena en dos mitades a partir de la línea de interés, comenzando la grabación desde uno de los lados. A partir de este momento, todos los planos siguientes serán filmados desde el mismo lado, independientemente de la posición de la cámara, sin permitir cruzar la línea al otro lado, tal y como se aprecia en la Figura 2.7. Esta regla se basa en la perspectiva de las imágenes grabadas y en la noción de profundidad inmersa en ellas, obteniendo de esta manera una sensación de que los personajes están enfrentados, y evitando posibles desorientaciones del espectador.

Según el número de actores que intervienen en la escena, se pueden utilizar unas aproximaciones u otras, existiendo un amplio abanico de posibilidades y técnicas. Aquí se muestran algunas de las comunes.

- **Dos actores:** Para el caso de dos actores, la regla se basa en el principio explicado anteriormente (ver sección 2.6), pudiendo posicionar la cámara desde situaciones externas, internas, etc. Normalmente, las posiciones *over the shoulder* son las más utilizadas, intercaladas con primeros planos de posiciones internas para momentos dramáticos.
- **Tres actores:** En este caso, se suele hacer uso de las técnicas de dos personas, generando dos líneas de visión entre los pares de actores e

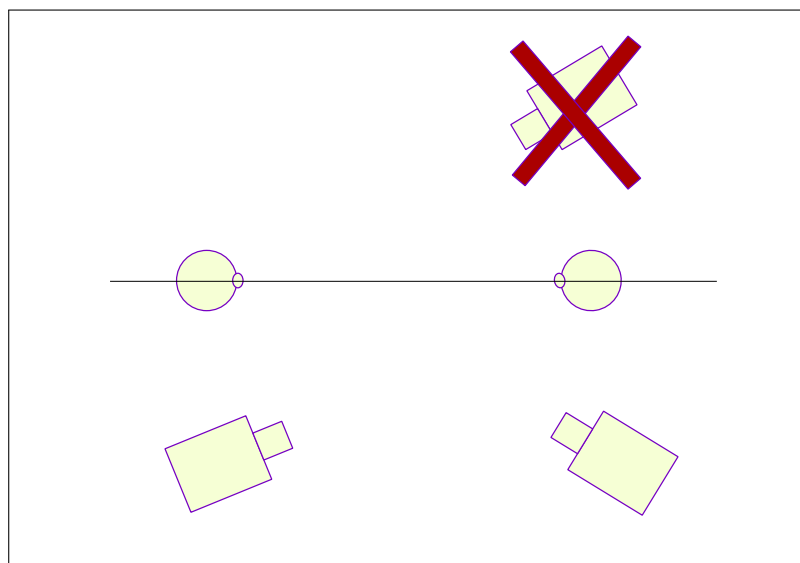


Figura 2.7: Regla de los 180°

interpretando las distintas posiciones de los personajes como una I, una L o un triángulo. Este tipo de tomas se suelen intercalar con planos largos desde posiciones Apex, para reorientar al espectador.

- **Cuatro actores o más:** En este tipo de escenas existen muchas posibilidades, pero por regla general se suele agrupar a los diferentes actores en conjuntos de 2 o 3 personas, haciendo uso de las reglas específicas para esas configuraciones en cada grupo.

## 2.8. Movimientos

Probablemente una de las técnicas que más dinamismo y realismo ha proporcionado al mundo del cine haya sido el movimiento de la cámara. En sus inicios, el único elemento del que se disponía para rodar una escena consistía en sucesiones de planos estáticos (ver sección 2.3), donde el movimiento venía de la mano de los actores y elementos de la escena, y que representaba una visión muy teatral de las películas. Con el paso de los años, se fueron introduciendo diferentes tipos de movimientos que permitían sentir al espectador la grata sensación, casi física, de adentrarse en las imágenes y viajar por ellas.

Actualmente se suelen identificar, en la gran mayoría de textos relacionados, un conjunto de movimientos básicos que abarcan casi la totalidad de los movimientos utilizados a lo largo de los años, siendo estos últimos variaciones y combinaciones de los básicos. Uno de los principales problemas que se suelen dar con los movimientos de cámara es la sobreexplotación de este

recurso, lo que produce gran desorientación al espectador. A continuación se va a mostrar una descripción detallada de estos movimientos y su correcta utilización, y cuya representación visual se puede ver en las Figuras 2.8, 2.9, 2.10 y 2.11.

- **Panorámica Horizontal o *Pan*:** Como su propio nombre indica, este tipo de movimiento fue utilizado en sus orígenes para representar en pantalla una visión completa de escenas distantes, como por ejemplo, unas montañas, mediante un movimiento lateral de la cámara a lo largo de todo el espacio visual. Actualmente, el movimiento *Pan* se utiliza para cualquier toma en la que la cámara necesite moverse de una posición a otra para seguir la acción que se está desarrollando en la escena. Existen muchos motivos para utilizar este tipo de movimiento, pero el más importante involucra el movimiento de algún elemento importante en la escena, y que tiene que ser visualizado por el espectador, como la entrada o salida de un actor o el movimiento de éste a través de una habitación.

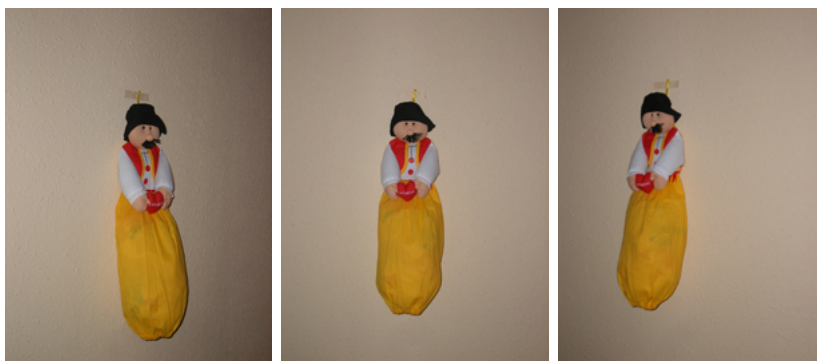


Figura 2.8: Panorámica Horizontal o *Pan*

- **Panorámica Vertical o *Tilt*:** Al igual que el *Pan*, la cámara puede producir un movimiento a lo largo del eje vertical de visión, de arriba a abajo o viceversa. Este tipo de tomas se suelen utilizar para mostrar situaciones en las que uno de los elementos realiza una acción, o movimiento, que implica un cambio vertical en la línea de visión de la cámara, como por ejemplo, un personaje que se levanta de una silla. Otro uso muy aceptado de este movimiento se basa en enfatizar la grandeza de elementos de gran tamaño, como por ejemplo un rascacielos, o disminuir la importancia de los de tamaño reducido, como por ejemplo un animal.

Figura 2.9: Panorámica Vertical o *Tilt*

- **Zoom:** En realidad esta técnica no es un movimiento propiamente dicho, ya que la cámara está estática y es la lente la que produce un acercamiento al objeto, pero suele ser considerado como tal por su efecto. La principal motivación de este movimiento consiste en aumentar la importancia o dramatismo del objeto sobre el que se realiza el *Zoom*, a la vez que supone un cambio suave en la escena, al contrario que el cambio entre un plano largo y uno corto.

Figura 2.10: *Zoom*

- **Dolly:** También conocido como *Track*, produce la misma sensación que el *Zoom*, pero en este caso sí conlleva un movimiento de la cámara, acercándose o alejándose del objeto.

Figura 2.11: *Dolly*

Esta clasificación no limita los movimientos de las cámaras a estos cuatro tipos, pero muestra un esquema claro de las distintas posibilidades y combinaciones que se pueden dar en una película para enriquecer el aspecto visual de las acciones de las escenas.





---

## Capítulo 3

# Trabajo previo

A lo largo de este capítulo se mostrarán las diferentes investigaciones en las que se ha inspirado y apoyado este trabajo. Esta recopilación del estado del arte se divide en tres secciones, en las que se muestran las aproximaciones más relevantes en las que se basa esta memoria. En la primera sección se realizará una breve introducción a los entornos virtuales y a sus diferentes definiciones y características según el campo de aplicación. En la segunda sección se continuará con un examen de las diferentes técnicas, existentes hasta la fecha, para la gestión automática de cámaras, mostrando primero una visión general del campo y centrándose después en aquellas que se basan en la utilización de técnicas cinematográficas. En la última sección se presentará una revisión de las distintas técnicas de reconocimiento de emociones, así como de las teorías psicológicas en las que están inspiradas, mostrando finalmente un análisis de la aplicación **Emotag** utilizada en el sistema.

### 3.1. Los entornos virtuales

Existen muchas definiciones en la literatura sobre qué es un entorno virtual y qué características debe poseer, variando desde las muy restrictivas a aquellas de un ámbito más general (Ibáñez-Martínez, Delgado-Mata, y Aylett, 2006). Esta gran diversidad de opiniones respecto a la definición de los entornos virtuales se debe, principalmente, a la gran variedad de contextos en los que este tipo de aplicaciones han sido utilizadas, como por ejemplo, el entretenimiento, la arquitectura, la medicina, el ámbito militar, la educación o la robótica, por mencionar algunos.

Las definiciones más restrictivas de los entornos virtuales los clasifican como programas software de representación de un espacio físico, real o imaginario, en tres dimensiones (3D) y en tiempo real, en el que los usuarios pueden interactuar con él mediante sistemas tridimensionales de entrada y salida. En un ámbito más general, otras definiciones plantean los entornos virtuales simplemente como representaciones en 3D que pretenden simular

espacios y situaciones reales o imaginarias.

Desde una perspectiva más tradicional, y fuertemente influenciada por los juegos de entretenimiento, los entornos virtuales son sistemas en los que el usuario puede navegar por el espacio tridimensional mediante personajes, ya sean de carácter humano o no, conocidos como *avatares*. En este tipo de sistemas, cada día se hace más importante no sólo el hecho de la existencia de dichos personajes, sino de toda una población y entorno, controlados ambos por el propio sistema, con los que el usuario pueda interactuar.

Es por esto que, a día de hoy, muchos de estos sistemas son conocidos como Entornos Virtuales Inteligentes (Aylett y Cavazza, 2001), debido a que cada vez son más demandados una interacción y un comportamiento del sistema más humano, por lo que se han ido introduciendo a lo largo de los años diversas técnicas de Inteligencia Artificial que permiten dotar a los entornos, y a los elementos controlados por éstos, de una interfaz más interactiva y real capaz de responder a las diferentes reacciones de los usuarios a la vez que se les guía y orienta por el entorno.

El extremo de este tipo de sistemas, y que aún a día de hoy la tecnología no permite alcanzar, se la denominada Realidad Virtual, en donde el usuario es el propio avatar y las diferentes reacciones de este son interpretadas por el entorno mediante sistemas de entrada y salida específicos. Aunque algunos consideran ciertos sistemas predecesores de la Realidad Virtual, como cascos específicos que hacen que el usuario visualice el mundo virtual como si estuviera en él o sistemas de sensores capaces de identificar cierto tipo de movimientos humanos, lo cierto es que actualmente no se ha alcanzado esta meta.

### 3.1.1. Breve historia de los entornos virtuales

La evolución de los sistemas de entornos virtuales ha sido extraordinaria y muy amplia en los últimos años, ramificándose en gran diversidad de áreas y materias. En esta sección se introduce una pequeña síntesis de los inicios de este tipo de sistemas, que conllevan avances desde dispositivos de entrada y salida hasta sistemas de gráficos en 3D.

Es ampliamente aceptado que los primeros sistemas de entornos virtuales fueron desarrollados para uso militar, y fundamentalmente para sistemas de simulación de vuelo (Sbretchs et al., 2002) durante la década de los 70. Sin embargo, Ivan Sutherland (1963), científico especialista en gráficos por ordenador, es considerado como la primera persona en desarrollar una simulación en tiempo real de gráficos interactivos para un ordenador en 1963. Un poco posterior, aceptado en mucha literatura como realidad virtual, fue el desarrollo en 1970, por parte de Scott Fisher, de un sistema capaz de combinar gráficos de ordenador con imágenes estereoscópicas y sonido en tres dimensiones, con el que se podía interactuar mediante un casco y unos guantes virtuales.

En 1985 se desarrolló en el *NASA-Ames Research Center* el sistema VIEW (*Virtual Interactive Environment Workstation*) (Fisher et al., 1986), supervisado por Scott Fisher, el cual disponía de sistemas de reconocimiento de voz, sistemas de sonido en 3D, guantes y casco virtuales, y que permitía una experiencia mucho más realista del entorno virtual. En 1989 **Atari** lanzó al mercado el primer videojuego en 3D.

Otro de los grandes impulsos en este campo fue la aparición de los motores gráficos en 3D, o *game engine*, durante los años noventa de mano de los sistemas de videojuegos, los cuales supusieron un gran avance para este tipo de entornos, permitiendo gran difusión en múltiples áreas, tanto de uso comercial como científico, y consiguiendo en los últimos años una gran calidad.

A día de hoy, existe gran diversidad de entornos, dependiendo de la funcionalidad requerida en cada campo, como ya se ha expuesto, pero quizá una de las líneas que más peso está adquiriendo son los sistemas que unifican la Inteligencia Artificial con los entornos virtuales, y que como ya se ha dicho, se denominan Entornos Virtuales Inteligentes.

### 3.1.2. Aplicaciones de los entornos virtuales

Los entornos virtuales han sido utilizados en multitud de disciplinas a lo largo de los años, lo que ha propiciado que las características de éstos varíen según su uso y funcionalidad, pudiendo interactuar con los usuario o no, aceptando la existencia de avatares o no, haciendo uso de técnicas de Inteligencia Artificial o no, etc. A lo largo de esta sección se mostrarán algunas de las características de estos sistemas según su campo de aplicación, así como algunos ejemplos de entornos.

Como se ha dicho, la simulación de vuelo, (Robinson y Mania, 2007), fue uno de los primeros campos en ver la potencialidad de los entornos virtuales. En este caso concreto, los sistemas no se basan tanto en la calidad gráfica o la interacción con el entorno, como en la correcta simulación de las distintas variables que entran en juego en el pilotaje del avión, así como en su correcto manejo de los sistemas de entrada y salida. Además este tipo de entornos requieren de un perfecto funcionamiento de sistemas de coordenadas que puedan ubicar correctamente al objeto y la visualización de la tierra, así como de la física inmersa en todos los procesos de control del vehículo, existiendo en el caso de grandes simuladores todo tipo de controles de mando y recreaciones con asientos y cabinas móviles. Algunos ejemplos de software de simulación de vuelo actuales son el creado por **Google** para **Google Earth** o el simulador de vuelo *Flight Simulator X* de **Microsoft**.

La arquitectura, (Kahkonen y Whyte, 2003), ha sido otra de las áreas en las que los entornos virtuales han tenido gran aceptación, permitiendo generar simulaciones de futuras construcciones en las que se pueda navegar mediante lo que se conoce como *paseos virtuales*, y que proporcionan una

visión de lo que será la futura construcción. Estos sistemas no precisan tanto de un correcto sistema de física y una gran calidad en la interpretación de los sistemas de entrada y salida como los simuladores de vuelo, pero sí de una gran calidad visual casi realista y una amplia gama de perspectivas.

El área de la educación (Dillenbourg, Schneider, y Synteta, 2007) es otro de los campos que en los últimos años ha experimentado un gran avance en la adaptación de las metodologías educativas mediante sistemas de entornos virtuales. En estas aplicaciones no es tan necesaria una gran calidad visual como en la arquitectura, pero sí una interfaz fácil de utilizar y atractiva para el usuario. Además, este tipo de entornos suelen utilizar sistemas de multiusuario que permiten la interacción de varios alumnos al mismo tiempo en la resolución de los problemas. Otra de las figuras importantes en estos entornos suele ser el papel del tutor, ya sea como avatar o como ente imaginario, que va guiando a los alumnos por los distintos temas de aprendizaje. Existen multitud de sistemas de estas características, desde simulaciones de situaciones críticas a aulas virtuales.

El mundo del entretenimiento, de la mano de los videojuegos principalmente (Torres-Guizar, 2006), es otro de los grandes beneficiados de este tipo de entornos, donde uno o varios usuarios pueden interactuar con diversos elementos del entorno para lograr superar misiones u objetivos propuestos por éste. En este caso, los sistemas de Inteligencia Artificial son imprescindibles para el correcto funcionamiento del entorno, así como para obtener una interfaz más atractiva al usuario. De igual manera, la calidad visual se ha convertido en uno de los requisitos imprescindibles, y que gracias a los avances técnicos, más ha mejorado en este tipo de entornos, produciendo mundos virtuales cada vez más realistas, que incluyen sistemas de física real y eventos atmosféricos. A su vez, un correcto y sencillo sistema de entrada y salida es necesario para que los usuarios puedan manejar los avatares del entorno, así como un sistema de cámaras que permita mostrarles lo que está sucediendo en cada instante. Otra característica muy demandada en los últimos años en este tipo de entornos es la posibilidad de que varios usuarios puedan interactuar a la vez en el mundo virtual, colaborando a la hora de resolver los problemas planteados. Son lo que se conoce como *Massively Multiplayer Online Role-Playing Games (MMORPGs)*, y precisan de sistemas de conexión multiusuario en red. Existen multitud de ejemplos de estos entornos, pero quizás algunos de los que más han impresionado en los últimos años sean el *World of Warcraft*, un sistema de multijugadores en red ambientado en un mundo de fantasía, o el *Assassin's Creed*, ambientado en las antiguas cruzadas.

Siguiendo en esta línea del entretenimiento, pero todavía en un proceso previo de investigación, se encuentran los entornos virtuales utilizados para contar cuentos, denominados como *Interactive Storytelling* (Peinado, 2006). Estos sistemas son similares a los videojuegos en cuanto a la tecnología utilizada, con la diferencia de que precisan de sistemas de narratología para

ir guiando al usuario a través de la historia. Algunos incluso no permiten una interacción por parte del usuario, y simplemente se limitan a mostrar los cuentos producidos por el sistema de narratología.

## 3.2. Sistemas automáticos de gestión de cámara

Los sistemas automáticos de gestión de cámaras comprenden muchas disciplinas en el ámbito de la computación, tales como “*path-planning*”, gestión de oclusión, modelado de sistemas de comunicación de alto nivel, cálculo de distancias, narratología, juegos, etc. Por este motivo, ha habido a lo largo de los años, muchos y muy diversos sistemas propuestos por parte de la comunidad científica para la resolución de este problema.

Si bien es cierto que en las diferentes propuestas el problema a resolver es el mismo, conseguir automatizar un sistema capaz de obtener la máxima expresividad e información mediante una secuencia de imágenes, a través de la gestión de un conjunto de cámaras, las técnicas y herramientas utilizadas distan mucho de un sistema a otro, abriendo un amplio abanico de posibles soluciones dependiendo de la naturaleza del sistema.

En el ámbito de la gestión automática de cámaras se podrían englobar no sólo los sistemas virtuales, en los que se emula el funcionamiento de una cámara, sino también las aplicaciones reales en las que se utilizan sistemas de visión, como por ejemplo robots, donde tareas como el correcto tratamiento de la información filmada y la gestión de las futuras capturas pueden ser críticas para el funcionamiento del sistema. Sin embargo este tipo de sistemas, aunque poseen características similares y pretenden resolver el mismo problema, o variaciones de éste, escapan al estudio de este trabajo (pudiendo perfectamente extenderse en una o varias tesis).

A lo largo de esta sección se describirán las distintas soluciones que se han propuesto a este problema, dependiendo de la naturaleza y del tipo de entradas y salidas deseadas, poniendo especial énfasis en los sistemas de gestión de cámaras en los entornos virtuales, y en especial, en las pocas pero importantes, propuestas que han intentado adaptar las técnicas cinematográficas a estos sistemas.

### 3.2.1. Aplicaciones y gestores de cámara

Para poder realizar un análisis más detallado de las distintas aproximaciones que se han utilizado a lo largo del tiempo en los sistemas automáticos de gestión de cámaras, es necesario, en primer lugar, conocer las diferentes aplicaciones en las que tiene cabida un sistema de este tipo, y sus características especiales.

Existen multitud de parámetros que pueden discriminar unos sistemas de otros, como por ejemplo, el tipo de entrada, el dominio de la aplicación, el sistema de representación de las cámaras virtuales, etc. Todo esto ha

motivado que haya surgido una gran proliferación en este tipo de sistemas a lo largo del tiempo como, por ejemplo, videojuegos, sistema de domótica, sistemas de modelado, sistemas geográficos interactivos, etc., en los que se capturar en secuencias de imágenes lo que esta sucediendo, ya sea para visualizarlo, o para trabajar con las imágenes virtualizadas.

Frente a esta diversidad de sistemas, y sus posibles soluciones, Marc Chriestie et al. proponen en su análisis (2005), y más extendido en (2006), una clasificación de los distintos tipos de aplicaciones atendiendo a las características de sus sistemas de gestión de cámara, englobándolas en los tres grupos siguientes.

- **Sistemas interactivos:** En este tipo de sistemas, la finalidad más importante consiste en lograr un correcto mapeo entre los diferentes parámetros que controlan las cámaras y el sistema de entrada del entorno virtual, ya sea un teclado, un ratón, ambos o dispositivos especiales.
- **Sistemas en reactivas en tiempo real:** En este tipo de sistemas se hace uso de técnicas similares a las utilizadas en robótica y gestión de sensores.
- **Sistemas de carácter general:** Este grupo comprende el conjunto de aplicaciones que están surgiendo actualmente, y cuya finalidad es dotar de un control de más alto nivel mediante la caracterización lógica de los parámetros de la cámara y el camino a seguir por ésta.

Según esta clasificación, Marc Chriestie et al. proponen cuatro categorías en las que se encuadran las distintas técnicas para la resolución de la gestión de cámaras en relación a la naturaleza del sistema y al dominio de la aplicación. Así, dividen las diferentes aproximaciones en los siguientes grupos:

- **Aproximaciones interactivas:** Gestionan el funcionamiento de las cámaras en respuesta a las directrices de un usuario mediante los dispositivos de entrada.
- **Aproximaciones reactivas en tiempo real:** Gestionan el control de cámaras mediante técnicas basadas en sistemas de robótica.
- **Aproximaciones algebraicas:** Representan el problema y la solución mediante sistemas de vectores.
- **Aproximaciones optimizadas y basadas en restricciones:** Estos sistemas modelan las propiedades de las cámaras mediante restricciones y funciones objetivo.

A continuación se mostrarán los trabajos más relevantes realizados hasta la fecha en las distintas categorías, mostrando la gran variedad de técnicas y planteamientos que se han propuesto por parte de la comunidad científica, y centrándose en los aspectos más importantes de cada uno.

### Técnicas en los sistemas interactivos

Como ya se ha adelantado, el aspecto fundamental en el control de la cámara en estos sistemas se basa en la manejabilidad y expresividad que se puede obtener mediante el mapeo de los diferentes parámetros de la cámara con los dispositivos de entrada del sistema. De esta manera, se consigue un sistema que permite al usuario decidir, mediante los movimientos de las cámaras, qué elementos del entorno virtual van a ser mostrados.

Las primeras investigaciones en este tipo de sistemas basaron sus esfuerzos en obtener modelos capaces de representar correctamente los tres grados de libertad en las rotaciones y su correcto mapeo con las diferentes entradas, ya que los sistemas de traslación no supusieron problema alguno. Diversas aproximaciones siguieron esta línea de investigación, como por ejemplo, los sistemas que hacen uso de tabletas gráficas para gestionar las entradas (Evans, Tanner, y Wein, 1981), las técnicas basadas en la selección del plano de trabajo y en la realización de los movimientos en los otros dos planos (Nielson y Dan R. Olsen, 1987) y, más modernos y similares a las aplicaciones actuales, los sistemas basados en la idea de envolver al objeto movable en una esfera lógica para gestionar así su control mediante algoritmos matemáticos, (Chen, Mountford, y Sellen, 1988; Hanson, 1992; Shoemake, 1992; Shoemake, 1994).

Otro de los aspectos fundamentales en los sistemas interactivos son las diferentes posibilidades de representar la cámara de acuerdo a los distintos modos de exploración del entorno virtual. Ware y Osborne propusieron en (Ware y Osborne, 1990) tres tipos de representaciones de lo que ellos llamaron “metáforas de control de cámara”, en las que se recoge un amplio rango de las aproximaciones existentes:

- *Camera in hand*: Estos sistemas se basan en interpretar la cámara como si estuviera en la mano del usuario, representando las rotaciones y movimientos que éste realiza. Un ejemplo de este tipo de representación se puede ver en el sistema de Brooks (1987).
- *World in hand*: Se caracteriza por pivotar la cámara alrededor del mundo mientras el foco apunta a una dirección, así un movimiento a la izquierda del usuario realiza un desplazamiento a la derecha en el entorno virtual. Un ejemplo de esta representación fue creado por Phillips en su sistema *Jack*, (1992).
- *Flying vehicle*: En este último tipo de representación, la cámara se

trata como si fuera un vehículo móvil a cierta altura, como si estuviera situada en un avión. Un ejemplo de este tipo de representación se puede ver en los sistemas de Brooks (1987; 1988), en los que se utilizan joysticks para el control de velocidad.

Otra aproximación, en esta línea, para la representación de la cámara, es la denominada *walking metaphor*, en la cual la cámara se sitúa a una distancia fija del suelo, intentando de esa manera evitar posibles colisiones con las superficies, (Hanson y Wernert, 1997). Burtnyk et al. (2002) proponen una técnica similar, en la que la cámara está situada a una distancia fija de la superficie que rodea al objeto explorador, restringiendo las posiciones de ésta a determinados puntos de interés en el espacio virtual, basándose en (Phillips, Badler, y Granieri, 1992), y garantizando cierta calidad en la exploración. Profundizando más en esta idea, Khan et al. (2005) proponen un sistema de exploración en el que la cámara se mantiene a una distancia fija del objeto y de las distintas superficies del entorno, evitando así la oclusión y colisión.

Otro tipo de técnicas completamente diferentes, pero que también pretenden evitar las colisiones y la oclusión entre la cámara y los objetos del entorno, se basan en el estudio de modelos físicos para el control de cámara, incorporando nociones como la inercia o la fricción y considerando las entradas del usuario como fuerzas que actúan sobre la cámara, (Turner et al., 1991). Parte de esta teoría fue utilizada por Hanson et al. en (1997) y Xiao y Hubbold en (1998), donde presentan sendos trabajos basados en sistemas de gravedad para evitar colisiones.

La mayoría de estas técnicas se centran en las distintas representaciones para explorar el entorno virtual, pero no inciden en las posibilidades que surgen al cambiar de una a otra “metáfora”. Drucker et al. fueron unos de los primeros en estudiar este tipo de transiciones en su sistema de cámaras **CINEMA** (1992), en el que se combinan las distintas metáforas, permitiendo incluso al usuario añadir nuevas. Un ejemplo de la utilidad de este tipo de aproximaciones se puede ver en (Zelevnik y Forsberg, 1999), donde se muestran transiciones suaves entre los distintos modos de representación. Más centrado en los movimientos de la cámara, pero basándose en estos conceptos, es el trabajo de Mackinlay en el que se presenta un sistema de movimientos de cámara que varían según la distancia al objeto, (1991).

Más ejemplos y aproximaciones más recientes del estudio de las distintas metáforas y las transiciones entre ellas se pueden apreciar en los trabajos de Jung et al. (1998), Tan et al. (2000) o Li y Hsi (2004).

### Técnicas en los sistemas reactivos

Los sistemas reactivos hacen uso de las propiedades de las imágenes mostradas para extraer la información necesaria, y realizar una gestión de



cámara suave, continua y evitando la oclusión entre el objeto principal y la cámara. Para realizar todos estos ajustes se basan, como se ha dicho, en nociones de robótica y planificación mediante sensores. Un ejemplo claro de este tipo de técnicas son las utilizadas en las diferentes aproximaciones de *path-planning*, o planificación de caminos, tarea muy común en los entornos virtuales en los que, en muchas circunstancias, la cámara debe seguir a elementos móviles.

Este grupo de técnicas son muy similares a las utilizadas en las aproximaciones de control visual o *visual servoing*, en las que se define una tarea u objetivo que será alcanzado mediante el análisis de las propiedades y características de las imágenes del entorno. Para ello, se considera el sistema de visión como un sensor más dedicado a esa tarea, y encapsulado en un bucle de control. Atendiendo a esto, se define el objetivo o función que representa el problema, como el posicionamiento de la cámara, y se asocian unas restricciones a esta función, por ejemplo, una determinada distancia al objeto principal y la localización de éste en el centro de la escena filmada, obteniendo así la forma de calcular la posición adecuada a partir de la información proporcionada en las imágenes. Una descripción más detallada de este tipo de técnicas se puede consultar en (Hutchinson, Hager, y Corke, 1996).

Estas nociones han sido abstraídas a los sistemas de cámaras para entornos virtuales, en los que no sólo hay que restringirse a imágenes captadas por sensores, ya que se posee toda la información de la escena a través del mundo virtual. De esta manera, han surgido diversas aproximaciones que se han servido de estas ideas para gestionar la trayectoria de las cámaras persiguiendo a elementos móviles, asociando restricciones a ésta y basándose en funciones matemáticas que hacen uso de parámetros como la velocidad, el movimiento del objeto a seguir o la distancia con éste (Courty y Marchand, 2001). Otra técnica muy común, que se rige por estos mismos principios, consiste en evitar los diferentes obstáculos que pueden aparecer en la trayectoria de una cámara mediante la minimización de funciones y el análisis de la imagen que se está representando, como por ejemplo, calculando la función de coste de la distancia entre el objeto que se interpone y la cámara, (Marchand y Courty, 2002; Marchand y Hager, ).

Este tipo de aproximaciones son muy eficientes computacionalmente y muy versátiles, propiedad muy interesante en entornos dinámicos como los videojuegos, en donde es fácil obtener las reacciones de los personajes y generar las restricciones de las cámaras respecto a ellos. Siguiendo esta línea, Halper et al. proponen en (2001) un sistema de cámaras donde las posiciones y ángulos se calculan a partir de eventos generados por los personajes, a los que se les asocia propiedades como el ángulo de interés, posicionamiento en la pantalla, etc.

En general, este tipo de técnicas se basan en la predicción de futuras situaciones, atendiendo a características similares de hechos ya pasados para ajustar los parámetros de la cámara adecuadamente.

### Técnicas en los sistemas algebraicos

Una de las primeras aproximaciones de los sistemas algebraicos se puede ver en el trabajo de Blinn's (1988), donde se presenta un sistema de gestión de cámara automático que permite visualizar en la pantalla, con las coordenadas y tamaños dados, la sonda y el planeta que está atravesando. Las técnicas algebraicas basan el posicionamiento de la cámara en la resolución de sistemas de vectores definidos en el modelo del mundo que se está mostrando, en el que los objetos están definidos como puntos, con la ventaja de la eficiencia computacional en el cálculo de la solución y la restricción de ésta a las dimensiones de la pantalla.

La forma óptima de obtener la solución en estos sistemas se basa en la resolución de la ecuación 3.1, que expresa la transformación de un punto  $P(x, y, z)$ , representado en las coordenadas espaciales del mundo virtual, a las coordenadas  $S(x', y')$  de la pantalla donde se tiene que visualizar. Este proceso se realiza para cada punto del entorno multiplicando éste por la matriz  $H$ , donde  $P$  es la matriz de proyección,  $R$  es la matriz de rotación y  $T$  es la matriz de traslación.

$$S \begin{pmatrix} x' \\ y' \end{pmatrix} = P * R * T * P \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} = H * P \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} \quad (3.1)$$

Esta fórmula representa las variaciones de los objetos en el espacio, obteniendo las coordenadas en la pantalla, que serán las entradas de los sistemas algebraicos a resolver, junto con sus posiciones en el espacio en tres dimensiones, el *up-vector* y la apertura de la cámara. Estudiando la posición de la cámara en el espacio, y la representación en la pantalla de los objetos, se infieren, a través de las fórmulas, las nuevas posiciones de ésta.

Otro campo importante son las aplicaciones basadas en sistemas de dos dimensiones, las cuales han hecho uso de los métodos algebraicos para resolver problemas como la generación de imágenes de fondo, a partir de directrices en tres dimensiones de los distintos movimientos de la cámara, (Wood et al., 1997), o más actual, para gestionar los movimientos y posiciones de la cámara en los sistemas de guía multimedia de museos, en los que se hace énfasis sobre elementos de cuadros o frescos a la vez que el audio describe los mismos, (Zancanaro, Rocchi, y Stock, 2003).

### Técnicas en los sistemas de optimización y basados en restricciones

Estos sistemas hacen uso del estudio de las diferentes propiedades de las cámaras, sus proyecciones en la pantalla y sus movimientos, para definir funciones o restricciones mediante las que inferir el siguiente movimiento o posición. Para lograr esto las diferentes aproximaciones se basan en la

expresividad y riqueza de las distintas técnicas cinematográficas, expresando las escenas mediante propiedades para, en un análisis posterior, extraer las relaciones que permitirán deducir nuevas configuraciones de cámara.

Atendiendo a la generación de estas relaciones, se pueden clasificar las distintas aplicaciones según las técnicas utilizadas, diferenciando entre sistemas basados en restricciones y sistemas de optimización. En los primeros, el principal objetivo es satisfacer un conjunto de restricciones sobre las propiedades de la cámara, las cuales se pueden relajar si, por ejemplo, ocurriera un problema con la configuración obtenida, como la oclusión con un objeto, o no existiera solución. El segundo tipo de sistemas se basa en la maximización, o minimización, de funciones cuyos parámetros son las propiedades de la cámara.

Las primeras aproximaciones de los sistemas basados en restricciones se centraron en la resolución del problema de *path-planning*, o planificación de caminos, utilizando para ello restricciones sobre las propiedades de la nueva toma deseada, como el tamaño de los objetos o las proyecciones de las imágenes, (Jardillier y Languénou, 1998). Otra rama de este tipo de sistemas basa sus técnicas en la resolución de intervalos aritméticos mediante los que se representan todas las posibles soluciones. De esta manera se comprueba la posible solución a un problema, que es representado como un intervalo en un dominio de acuerdo con las restricciones, evaluando si existe solución, no existe o se desconoce. El principal problema de estas aproximaciones radica en el alto coste computacional, además de existir la posibilidad de no encontrar solución. Un ejemplo de esto se puede ver en el trabajo de Christie et al., (2002), donde las restricciones están basadas en técnicas cinematográficas y los movimientos están encapsulados en elementos denominados *hypertubes*. Un análisis detallado puede encontrarse en el artículo (Benhamou et al., 2004). Otros sistemas, menos generales, como (Bares, Grégoire, y Lester, 1998) permiten relajar automáticamente las restricciones mediante una jerarquía para evitar la posibilidad de que el sistema no encuentre solución. Sistemas más actuales como (Bourne y Sattar, 2005) y (Bourne, Sattar, y Goodwin, 2008) hacen uso de estas técnicas para la gestión de cámaras en entornos interactivos en tiempo real.

Los sistemas de optimización, como se ha dicho anteriormente, basan sus técnicas en la maximización de funciones que representan las características deseadas de la cámara, parametrizando estas funciones con las propiedades de ésta. Existe un amplio abanico de técnicas que permiten optimizar las funciones objetivo, como por ejemplo sistemas deterministas, algoritmos genéticos o métodos probabilísticos. El problema a resolver en este caso es matemáticamente expresable como la búsqueda de la configuración de cámara más óptima dentro del conjunto de todas las configuraciones posibles. Un ejemplo de estos sistemas se puede ver en (Halper y Olivier, 2000; Pickering, 2002).

Si bien es cierto que ambas técnicas suponen una buena aproximación

a la resolución de la gestión de sistemas automáticos de cámaras, lo cierto es que los mejores resultados han venido de la mano de sistemas que combinan las bondades de ambas. De esta manera se han presentado diversos sistemas que modelan las diferentes propiedades de las cámaras mediante un conjunto de restricciones y un conjunto de funciones de optimización, o funciones objetivo, obteniendo los beneficios de ambos y minimizando el coste de ejecución (Drucker y Zeltzer, 1995). Aún con esto, la carga computacional de estas técnicas supone el principal cuello de botella en este tipo de aplicaciones, por lo que muchas aproximaciones han utilizado diferentes operadores geométricos, basados en las posiciones de la cámara o mediante heurísticas, para discriminar y reducir el espacio de búsqueda de las técnicas basadas en restricciones, minimizando así el tiempo de ejecución. Ejemplo de esto se puede ver en (Bares et al., 2000; Bares, Thainimit, y McDermott, 2000; Pickering, 2002; Christie y Normand, 2005).

### 3.2.2. Aplicación de técnicas cinematográficas a la gestión de cámara

Como se ha dicho, existen muchas y muy diversas técnicas que permiten automatizar los sistemas de gestión de cámaras en los distintos campos de la computación gráfica. Sin embargo, por regla general, la gran mayoría no incide en las características que aportan las distintas nociones, que a lo largo de todo el siglo XX, el cine ha desarrollado y utilizado. Características como la expresividad, suavidad de movimientos en las transiciones entre secuencias o el posible impacto sobre el espectador, conseguirían un mayor realismo en este tipo de aplicaciones, enriqueciéndolas en gran medida. Por todo ello, en los últimos años han surgido diferentes trabajos que han intentado abstraer ese conocimiento, modelizándolo según las distintas técnicas expuestas en el apartado anterior, y resolviendo, según las características del sistema y de la aproximación utilizada, los problemas que se plantean al pasar del mundo abstracto de las técnicas cinematográficas al mundo geométrico y matemático de los entornos virtuales.

A lo largo de esta sección se mostrarán los sistemas más importantes, hasta la fecha, que se han servido de las nociones de las técnicas cinematográficas para intentar mejorar los sistemas de cámaras, no solo desde el punto de vista de la expresividad, si no también centrándose en las distintas formas de control de la cámara que se pueden adoptar al introducir los parámetros y las propiedades necesarias de estos principios.

Uno de los trabajos pioneros en incorporar estos conceptos a un gestor de cámaras de un entorno virtual fue el sistema presentado por Drucker y Zelter en (1994; 1995), donde la noción de módulo de cámara, elemento que abstrae y encapsula los controles y restricciones de la cámara en el *Framework*, y que representa un período de tiempo que comprende el inicio y finalización de la visualización de una escena en el que se preserva la continuidad, está basado

en el concepto cinematográfico de toma, que representa el período de tiempo de una secuencia en la que la cámara está en una posición estática, o el intervalo comprendido entre el inicio y finalización de un tipo de movimiento.

Aunque es cierto que este trabajo supuso el origen de este tipo de sistemas, no se puede hablar de su gestión de cámara como un módulo puramente cinematográfico, ya que simplemente se equiparan conceptualmente elementos del sistema con principios cinematográficos, sin basar el control de las cámaras en éstos.

Fueron Christianson et al. quienes en (1996) trataron de resolverlo modelizando las diversas técnicas del cine mediante un lenguaje declarativo al que llamaron *Declarative Camera Control Language* (DCCL), que formaliza mediante una gramática, el conocimiento que se puede extraer de los diferentes manuales de cine. Para realizar este lenguaje se basaron en el concepto de *film idioms* o lenguajes cinematográficos, y principalmente en la gran clasificación de técnicas que propone Arijon en (1991 (originally published 1976)). Estos lenguajes definen diferentes esquemas para filmar las distintas acciones que se pueden dar en una historia, como, por ejemplo, qué planos y movimientos de cámara son los más adecuados para una secuencia de diálogo entre dos personas.

DCCL abstrae los conceptos de *film idiom* y plano para representar el control de cámara de las distintas situaciones de una historia, e introduce la noción de *Fragment*, que representa un intervalo de tiempo en el cual la posición y orientación de la cámara es la misma. El *Fragment* está definido en DCCL como la primitiva del sistema, de tal manera que los *film idioms* están formados por un conjunto de planos, los cuales a su vez se componen de un conjunto de *Fragments*. Además, los *Fragments* encapsulan la información de control de la cámara que permite especificar parámetros típicos de las técnicas cinematográficas como la distancia de la cámara al objetivo, la posición de ésta, el tipo de movimiento, los actores que forman parte de la escena, etc. Un ejemplo de este lenguaje se puede ver en la Figura 3.1.

Para probar este lenguaje (1996) presentaron el sistema *Camera Planning System*. Esta aplicación permite generar una secuencia completa de cámaras a partir de una traza de animaciones, que representan las acciones que han ocurrido en una simulación previa de la animación, junto con el número de secuencias de la historia. El sistema se divide en tres módulos fundamentales:

- *The Sequence Planner*: Esta parte del sistema toma como entrada la traza de animaciones y genera un árbol que representa la estructura de una película con un nodo raíz denominado *film*, cuyos nodos hijos representan las secuencias, y de los que, a su vez, penden las escenas. En este módulo, a cada una de estas escenas se le asigna todos los posibles *film idioms* compatibles, para lo que se realiza un mapeo entre las acciones de la escena y las actividades típicas de cada tipo de

```

(AcFilmIdiom name Arijon-13-2
:parameter (AcParamApproach :actor1 :actor2 :start :stop)
:line (AcLineIdiom :primary ?actor1 :other ?actor2 :side left)
(AcFilmShot name shot1
  (AcFragGoBy name frag1
    :time ?start :primary-moment beginning :entry-pos center
    :exit-pos out-left
    :placement (AcPlaceInternal :primary ?actor1 :other ?actor2
      :range closeup :primary-side center)))
(AcFilmShot name shot2
  (AcFragGoBy name frag2
    :time ?frag3.first-tick :primary-moment end
    :entry-pos on-right :exit-pos center
    :placement (AcPlaceExternal :near ?actor1 :far ?actor2
      :primary-subject near :range longshot :primary-side
      center)))
(AcFilmShot name shot3
  (AcFragGoBy name frag3
    :time ?stop :primary-moment end :entry-pos out-right
    :exit-pos right12
    :placement (AcPlaceApex :primary ?actor1 :other ?actor2
      :range mediumshot :primary-side right12))))

```

Figura 3.1: Lenguaje DCCL correspondiente a una conversación entre dos personas.

lenguaje cinematográfico.

- *The DCCL Compiler*: En este caso, se parte del árbol creado en el módulo anterior y se expanden, mediante la información de movimientos de los actores en la escena, los *film idioms* en *frames*, que poseen las propiedades geométricas del entorno virtual.
- *The Heuristic Evaluator*: Mediante heurísticas basadas en técnicas cinematográficas, como por ejemplo, que las cámaras no crucen la línea de visión, crear movimientos suaves, etc., se selecciona la lista con los *frames* más apropiados para cada escena, generando así una secuencia completa para la gestión de la cámara.

Siguiendo esta línea de intentar obtener un mayor realismo y expresividad mediante técnicas de cine modelizadas en los diferentes *film idioms*, He et al. proponen en (1996) el sistema interactivo en tiempo real *The Virtual Cinematographer*. Esta aplicación es capaz de decidir las posiciones y movimientos de la cámara a partir de los eventos generados en el sistema virtual *Virtual Party*, en el que un grupo de personas pueden interactuar con sus

*avatares* mediante un conjunto de acciones predeterminadas.

El sistema *The Virtual Cinematographer* está formado por dos componentes principales, el *Camera Modules* y los *Idioms*, que representan los distintos módulos de cámara posibles y los lenguajes cinematográficos respectivamente. Las diferentes técnicas están agrupadas en los distintos *Idioms* creados para la aplicación, los cuales están implementados mediante una máquina de estados finitos, formada por los módulos de cámara que componen el lenguaje. Estos módulos de cámara representan las distintas operaciones básicas de cámara, como posiciones, movimientos, etc., que se pueden llevar a cabo en el entorno virtual. Las transiciones entre los estados se realizan atendiendo a los eventos del entorno virtual y las reglas implementadas en los distintos *Idioms*, que están organizados jerárquicamente pudiendo pasar de los más concretos a los más generales si no se dan los elementos necesarios para activar los más específicos. Así las transiciones se van produciendo dentro de los *Idioms*, hasta que se cumplan las propiedades que activen otro lenguaje.

El principal problema de las aproximaciones basadas en *film idioms*, ya sea mediante una gramática o mediante una máquina de estados, radica en la posibilidad de no encontrar solución a problemas que pueden surgir en tiempo real, como la interposición de un objeto entre la cámara y el objeto filmado o configuraciones espaciales no contempladas en los *film idioms*. Bares y Lester proponen en (1998; 2000) un sistema de gestión de cámara que trata de solucionar este tipo de situaciones, modelizando el conocimiento cinematográfico mediante restricciones en las configuraciones de cámara, pudiendo suavizarlas en caso de producirse, por ejemplo, un problema de oclusión.

Este sistema recibe como entrada, bien por un usuario o por un software específico, objetivos cinematográficos que el gestor de cámara debe satisfacer de acuerdo con la escena actual, como por ejemplo, el objetivo de mostrar un objeto que ha cogido un personaje en la escena, o seguir a un personaje a lo largo de una ciudad. Estos objetivos son analizados y formulados mediante restricciones basadas en parámetros de la cámara y en los objetos que componen la escena, como incluir al personaje en el campo de visión de la cámara, el ángulo de orientación de ésta, la distancia entre la cámara y el personaje o evitar la oclusión con los objetos del entorno. Estas nuevas restricciones generadas son evaluadas para determinar si existe una configuración de cámara que satisfaga, acorde con la escena actual, los objetivos. Si no fuera así, el sistema relajaría las restricciones hasta que una configuración fuera posible, o descompondría el problema en múltiples subproblemas, analizando si alguna de las subsoluciones sería plausible en la escena.

Actualmente, los distintos sistemas basados en restricciones hacen uso de un conjunto de técnicas de optimización para mejorar la eficiencia, basando sus propiedades como movimientos, posiciones, ángulos, etc., en los conceptos cinematográficos, al igual que los sistemas anteriores. Un ejemplo



de esto se puede ver en el trabajo propuesto por Christie et al. en (2002) y Christie y Languénou en (2003), donde el usuario proporciona los movimientos y propiedades que desea obtener del gestor de cámaras y el sistema, a través de los *Hypertube* y las restricciones, que representan los movimientos de las cámaras de cine y las propiedades cinematográficas respectivamente, instancia los objetivos descritos por el usuario.

Una aproximación basada en ambas técnicas se propone en el trabajo de Amerson y Kime en (2001). Este sistema abstrae el conocimiento de los distintos *film idioms* mediante un árbol organizado jerárquicamente denominado *scene tree*, de tal manera que la raíz del árbol representa el lenguaje que recoge las situaciones más generales, especializando éstas según se avanza por el árbol, hasta llegar a las hojas, que simbolizan las secuencias más específicas recogidas por los lenguajes cinematográficos. Cada *film idiom* está codificado mediante un conjunto de escenas en el lenguaje creado por Amerson y Kime *FILM (Film Idiom Language and Model)*, donde cada escena está formada, a la vez, por un conjunto de planos, que representan un movimiento simple de cámara, y tienen asociadas un conjunto de restricciones que pueden ser suavizadas en tiempo real.

La información necesaria para que el sistema pueda gestionar las cámara en tiempo real es proporcionada por un módulo de narrativa que recopila las distintas acciones de los personajes que componen la escena, y las consecuencias de éstas. Estos datos son procesados por el módulo *The Translator* y convertidos al formato interno que utiliza el módulo *The Director*, determinando en cada escena parámetros como el número de actores, el tipo de escena, el sentimiento, etc. Con esto, el módulo de dirección realiza una búsqueda en profundidad sin vuelta atrás en el *scene tree* para determinar cual es el *film idiom* más adecuado. En último lugar se realiza una comprobación de las restricciones y el entorno virtual para determinar, por ejemplo, que no existe oclusión. Si así fuera, el módulo *The Cinematographer* relajaría las restricciones hasta obtener el plano más adecuado, evitando así configuraciones erróneas.

Enfoques similares, como el propuesto por Halper et al. en (2001), utilizan este tipo de conceptos para la gestión de cámaras en videojuegos, donde un conjunto de plantillas representan el conocimiento cinematográfico asociado a acciones o situaciones determinadas. Así cada, *Shot Template* está compuesta por un conjunto de restricciones, que al igual que en los sistemas anteriores, se pueden suavizar en caso de producirse errores. Estas plantillas están formadas por restricciones basadas en sentimientos mediante las *Emotion Templates* y por un conjunto de plantillas llamadas *Shot Library*, que representan tipos de planos.

Este sistema no solo pretende mejorar los gestores de cámaras mediante la utilización de técnicas de cine, sino que además intenta que la transición entre las distintas secuencias sea suave. Para ello, hace uso de la entrada proporcionada por un módulo de narratología, junto con el estado actual



y previo de las cámaras, para predecir en tiempo real la siguiente configuración, persiguiendo llegar a un compromiso entre la coherencia de las imágenes y las restricciones.

Un sistema también basado en videojuegos, pero más centrado en el punto de vista de la interacción humana en tiempo real con el gestor de cámaras, se puede ver en el trabajo de Lin et al. en (2004). En este caso el conocimiento cinematográfico está encapsulado en los *Camera modules* y *Description Shots*, que son equivalentes a los *film idioms*, y están organizados jerárquicamente como en el sistema de Amerson y Kime, estructurados en una máquina de estados finita, donde cada estado es un *Camera module*. El sistema recibe de la aplicación en tiempo real la posición de los personajes, la orientación, la acción realizada, el equipamiento y las órdenes de entrada del usuario, que junto con el estado actual de cámara es utilizado para elegir la configuración de cámara más adecuada.

Otro tipo de sistemas utilizan las distintas interacciones entre las entidades que componen el sistema para determinar y modelizar el conocimiento cinematográfico mediante un sistema reactivo. Un ejemplo de esto se puede encontrar en el trabajo de Tomlinson et al., (2000), en el que un conjunto de agentes, que representan los personajes, interaccionan en el sistema virtual, y cuyo comportamiento y estado de ánimo influirán en las técnicas de cine a utilizar. La cámara es, también en este caso, un agente denominado *CameraCreature*, que determina el estilo con el que se mostrará el entorno virtual de acuerdo con las emociones del resto de entidades, que condicionaran su propio estado emotivo. Por ejemplo, si el estado es de felicidad, se producirán más cortes y movimientos de oscilación que si es un estado de tristeza.

En este caso, las nociones cinematográficas son abstraídas en objetos denominados *Shots*, que a su vez están formados por *elements*, mediante los que se representan las diferentes propiedades de las cámaras. Existen cuatro tipos de elementos en el sistema: sensores, emociones, motivaciones y acciones, cada uno de los cuales puede ser utilizados para inferir en la cámara un nuevo estado.

Un punto de vista más centrado en el aspecto dramático de las técnicas cinematográficas se puede ver en el trabajo presentado por Hornung et al. en (2003), en el que la cámara, representada como un agente autónomo en un sistema interactivo, elige la configuración más apropiada de acuerdo al contexto narrativo. Para ello, cada evento narrativo es descrito mediante ocho parámetros que definen la relevancia dramática del mismo, y de los que se extraen, junto con la información geométrica, el número de participantes, etc., los parámetros de las configuraciones de cámara mediante un sistema híbrido de reglas y redes neuronales.

Por regla general, la mayoría de estas aproximaciones se centran en el aspecto físico de las técnicas cinematográficas, sin tener en cuenta la capacidad expresiva y emotiva que hay inmersa en la mayoría de estas técnicas. Si

bien es cierto que en los últimos años han habido algunas aproximaciones en este campo, en todas ellas la información necesaria para poder gestionar este tipo de sistemas se obtiene por parte de un usuario o mediante configuraciones especiales asociadas a las distintas acciones que el sistema interactivo puede representar.

### 3.3. Sistemas de reconocimiento de emociones

Quizá uno de los aspectos más olvidados en los sistemas de interacción con humanos, y que en los últimos años está consiguiendo una creciente atención por parte de la comunidad científica, ha sido la capacidad de reconocimiento y transmisión de las emociones por parte de las distintas aplicaciones interactivas. La capacidad, tanto de identificación de emociones, ya sea en un texto o un sistema de voz, como de generación de éstas en las aplicaciones interactivas actuales, es una característica imprescindible que permite dotar de una capacidad más humana a estos sistemas, consiguiendo así una mejoría notable en la interacción entre el usuario y el sistema final.

Es por esto que actualmente están surgiendo numerosos campos de investigación en este ámbito, variando desde el reconocimiento de emociones en texto, en voz, o incluso en las propias expresiones faciales de los humanos, permitiendo así a los diferentes sistemas actuar de una determinada manera según el estado emocional analizado. De la misma forma, numerosas investigaciones están realizando grandes avances en el campo de la generación de emociones, tanto en los sistemas de generación del lenguaje natural como, en sistemas de síntesis de voz, o en la propia expresión facial de los robots humanoides, obteniendo así una interfaz más cercana al usuario.

Los sistemas de emociones no solo permiten una mejora en la comunicación con el usuario, sino que además, amplían el abanico de variables a la hora de elegir una determinada solución a un problema, permitiendo así una mejora en la respuesta del sistema. Como ya se ha dicho, las diferentes técnicas de dirección cinematográficas utilizadas tienen como uno de los componentes esenciales, a la hora de elegir las distintas posibilidades, la carga emocional que se desea transmitir en cada escena, y que viene inmersa en el guión de la historia. Es por esto que todo sistema de dirección automática basado en estas técnicas precisa de un sistema de reconocimiento de emociones, en este caso concreto, de identificación en el texto del guión, que permita obtener esta información para poder determinar la manera más apropiada de visualizar las escenas de la historia.

Todos estos sistemas se basan en diferentes teorías psicológicas expuestas a lo largo de los años sobre la identificación de los conceptos que representan el espacio afectivo de los seres humanos, así como en sus diferentes clasificaciones de las emociones existentes. A lo largo de la siguiente sección se realizará un análisis en profundidad de estas teorías y su evolución a lo

largo de los años. A continuación, se presentará, en las siguientes secciones, las diferentes técnicas de anotación semántica más utilizadas actualmente en los sistemas de Procesamiento del Lenguaje Natural, concluyendo con un análisis de la aplicación **Emotag** utilizada en este sistema.

### 3.3.1. La emoción

Uno de los principales problemas a los que se enfrenta este tipo de sistemas, a la hora de tratar la información, es la identificación de qué es una emoción y cómo se representa. A lo largo de los años, se han planteado muchas teorías psicológicas sobre qué entiende el ser humano por una emoción y cómo la interpreta, existiendo incluso a día de hoy gran diversidad de opiniones.

La identificación de los diferentes aspectos que entran en juego en los procesos de las emociones, así como la realización de una clasificación de éstas para su correcta identificación, ha sido de gran interés a lo largo de toda la historia de la humanidad, remontándose hasta los filósofos griegos. Actualmente existen dos teorías principales, ampliamente aceptadas, para representar y clasificar las diferentes emociones humanas: la teoría de dimensiones emocionales y la teoría de categorías emocionales. La primera de ellas se basa en la idea de que las emociones humanas se pueden representar mediante dos o tres dimensiones, variando conceptualmente según el autor, a través de las cuales se puede abarcar todo el espacio emocional humano. La teoría de las categorías emocionales, expone, por el contrario, que las distintas emociones humanas se pueden representar mediante un conjunto de palabras que denotan las emociones, en las que se engloba todo el espacio emocional humano. Las bases de estas dos líneas principales se pueden encontrar en los trabajos realizados por los psicólogos James Russel (1980) y William James (1884), respectivamente, los cuales han ido evolucionando con los años hasta las teorías actuales expuestas.

El trabajo de Russell se centra en la representación del espacio emocional humano mediante una estructura circular de dos dimensiones bipolares, el placer y la activación. Otro de los modelos emocionales en esta misma línea, fue el estudio realizado por Watson y Tellegen (1985), en el que se presenta una estructura circular emocional similar a la de Russell, pero en este caso basada en las dimensiones de afecto positivo y afecto negativo, la cuales a su vez poseen una intensidad baja o alta. Un ejemplo se puede ver en la Figura 3.2.

Inspirado en esta misma idea es el estudio realizado por Lang (1980), donde se expone el estándar SAM, a partir del cual se puede asignar a cada palabra inglesa un valor, generalmente entre 1 y 9, para cada una de las tres dimensiones propuestas en el estándar: evaluación, activación y control. Esta forma de asignación de los valores a las dimensiones está basada en la escala *Semantic Differential*, desarrollada en 1950 para el tratamiento

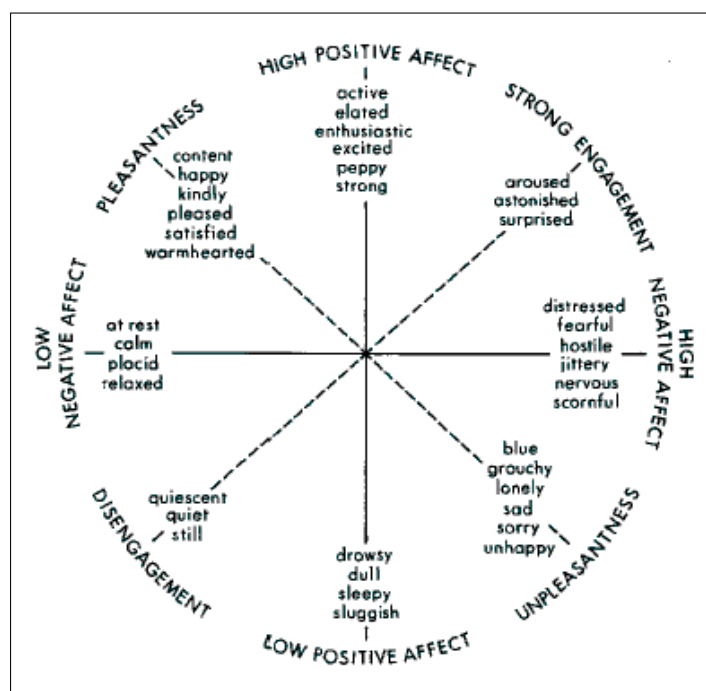


Figura 3.2: Mapa bidimensional de espacio emocional de Watson y Tellegen.

de emociones, cuya teoría se centra en la idea del pensamiento humano mediante polos opuestos, como el bien y el mal. En base a esto, Lang y Bradley (1999) crearon la *Affective Norms for English Words (ANEW)*, a través de la cual se pretende obtener un conjunto amplio de palabras medidas en términos de las tres dimensiones emocionales.

Por el contrario, (en la otra línea), James utiliza los conceptos de las emociones como entidades, como unidades discretas con límites entre ellas claramente diferenciados, que engloban todo el espacio emocional humano. Esta teoría ha ido evolucionando a lo largo de los años hacia las categorías emocionales, (Cowie y Cornelius, 2003), cuya idea se basa en la lógica interpretación de las categorías conceptuales de las emociones mediante las palabras para denotarlas en cada idioma.

Uno de los puntos claves de esta teoría es la gran diversidad de opiniones que existe sobre cuáles deben ser esas categorías, pudiendo variar desde unas pocas hasta infinitas (Ortony y Turner, 1990). Realmente, los orígenes de la idea de un conjunto de emociones primitivas se podrían remontar a René Descartes, (Anscombe y Geach, 1970), quien propuso la teoría de la existencia de un conjunto de emociones puras y primitivas a partir de las que se derivan el resto.

Más recientemente, Plutchik's (1980) propone un conjunto muy reducido de emociones básicas o primitivas (*anger, anticipation, disgust, joy, fear,*

*sadness* y *surprise*), afirmando que el resto de emociones son mezcla de éstas o derivadas. Ekman (1992) reduce todavía más el conjunto de emociones básicas, asumiendo que sólo existen seis emociones claramente diferenciadas para las expresiones faciales (*anger*, *disgust*, *fear*, *joy*, *sadness* y *surprise*). Izard (1977) determina que el conjunto básico de emociones está compuesto por: *anger*, *contempt*, *disgust*, *distress*, *fear*, *guilt*, *interest*, *joy*, *shame* y *surprise*.

En 1988 Ortony et al. (1988) presentaron el modelo OCC, el cual se ha ido estableciendo como estándar en la síntesis de voz, y que cuenta con 22 categorías emocionales: *pride* - *shame*, *admiration* - *reproach*, *happy* - *resentment*, *gloating* - *pity*, *hope* - *fear*, *joy* - *distress*, *satisfaction* - *fear-confirmed*, *relief* - *disappointment*, *gratification* - *remorse*, *gratitude* - *anger* y *love* - *hate*. Parrot (2001), por el contrario, presenta una lista mucho más extensa de emociones, pero jerarquizada en una estructura de árbol, en la que se encuentran las emociones básicas (*love*, *joy*, *surprise*, *anger*, *sadness* y *fear*), emociones secundarias y emociones terciarias.

Para poder calcular la intensidad emocional en este trabajo, se ha utilizado una fórmula basada en las categorías emocionales analizadas por el programa **Emotag**, donde a cada emoción se le ha asignado un peso, dependiendo de la carga negativa o positiva de la emoción, que puede variar entre el rango  $\{-1, 1\}$ . Para realizar esta asignación de pesos, este trabajo recoge parte de las ideas presentadas en el trabajo de Watson y Tellegen sobre el estudio afectivo, positivo y negativo, del espacio emocional humano. Todo este proceso se explicará con mayor detalle en la sección 4.3.2.

### 3.3.2. Técnicas en los sistemas de reconocimiento de emociones

Los sistemas de reconocimiento de emociones se pueden catalogar como un subconjunto dentro de los sistemas de anotación semántica, puesto que aunque no extraen un significado semántico como se suele entender en este campo, son capaces de identificar las emociones inmersas en un texto, aportando así un conocimiento, y en parte una clasificación, sobre el documento evaluado. Este tipo de sistemas, además, se basa en las mismas técnicas para la identificación de dichas emociones que los sistemas de anotación semántica, por lo que a lo largo de esta sección se realizará una breve introducción a las diferentes aproximaciones existentes.

Los sistemas automáticos de anotación son aquellos que realizan el proceso de clasificar o categorizar las palabras, frases, o expresiones de un texto, mediante entidades o significados semánticos. A través de estas técnicas se pretende solventar el cuello de botella producido por la adquisición de conocimiento anotado, que suele ser realizado a mano. Si bien es cierto que estos procesos no son completamente automáticos, ya que en todos los sistemas se precisa de la supervisión humana en algún punto, proporcionan una gran

mejora y un menor coste en los sistemas que precisan de este conocimiento.

Actualmente, se pueden distinguir dos líneas en este campo, atendiendo a los métodos utilizados: las basadas en patrones y las basadas en aprendizaje máquina. Esta distinción no implica la separación de ambas técnicas, y en muchos casos se utilizan los beneficios de ambas en lo que se denominan técnicas multi-estrategia. Un esquema de esto se puede ver en la Figura 3.3.

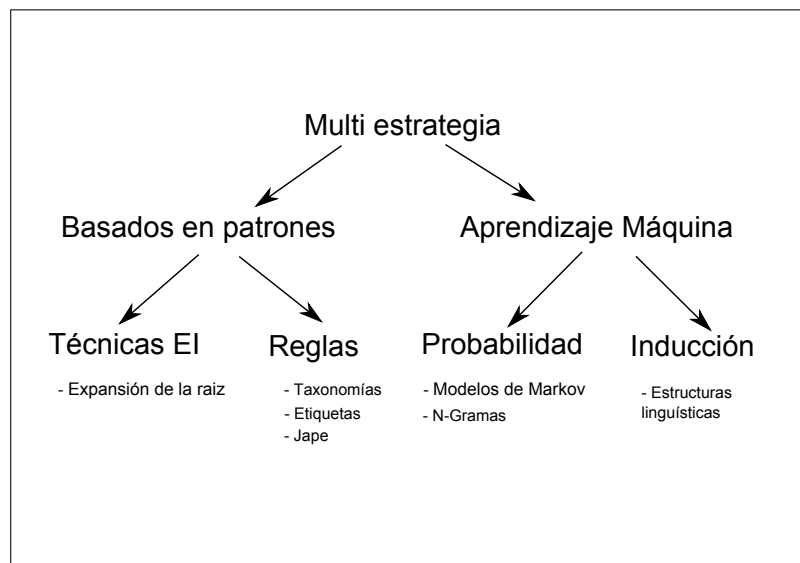


Figura 3.3: Clasificación de Sistemas de Anotación Semántica según sus métodos.

Los sistemas basados en patrones se suelen clasificar en patrones definidos manualmente mediante reglas o en patrones generados automáticamente mediante técnicas de Extracción de Información (EI). La mayoría de estos métodos se basan en el reconocimiento de patrones entre las entidades de un *corpus*, y para ello precisan de un conjunto inicial de entidades definido manualmente. El sistema propone nuevas entidades hasta que no encuentra ningún patrón más. Las aplicaciones basadas en patrones definidos manualmente suelen hacer uso de sistemas como JAPE (*Java Annotation Pattern Engine* (Cunningham, 1999)), sistemas de reglas, taxonomías o listas de palabras con categorías.

Los sistemas basados en aprendizaje máquina se dividen entre los que hacen usos de sistemas de modelos estadísticos para predecir la localización de las entidades en el texto, y los sistemas basados en inducción, determinada normalmente por la estructura de los textos y por características lingüísticas.

Otra de las principales características de este tipo de sistemas es la utilización de diferentes recursos semánticos de donde poder obtener el conocimiento necesario para realizar las distintas acciones llevadas a cabo en el proceso de anotación semántica. Es por esto que la mayoría de los siste-

mas hace uso de taxonomías, ontológicas, o repositorios semánticos como **WordNet**<sup>1</sup> o **Mikrokosmos**<sup>2</sup>, e incluso muchos sistemas se valen de los resultados propuestos por el buscador **Google**<sup>3</sup> a la hora de discriminar resultados. Además, en muchos casos se precisan de operaciones de EI previas para poder identificar distintos aspectos de los texto o palabras, por lo que se suelen herramientas como **GATE**<sup>4</sup>.

Se han desarrollado numerosas plataformas de anotación semántica en muy diversos ámbitos, la cuales incorporan, dependiendo de la implementación de cada una, las distintas técnicas expuestas anteriormente. Por ejemplo, AeroDAML (2001) hace uso de un sistema de patrones, especificados mediante reglas definidas manualmente, para la identificación de relaciones entre los sustantivos de un texto y la ontología DAML, utilizando como repositorio WordNet. KIM (2004) es un sistema más genérico de anotación basado también en patrones y en reglas definidas manualmente, valiéndose de una ontología propia denominada KIMO, en la que hay almacenadas más de 80.000 entidades relacionadas con localizaciones, organizaciones, etc. Armadillo (2004), por el contrario, realiza el proceso de anotación utilizando patrones generados automáticamente mediante técnicas de EI, ayudándose de los repositorios semánticos que desee proporcionar el usuario, y sin la necesidad de un *corpus* inicial. Ont-O-Mat (2002) también permite que el usuario sea quien proporcione el repositorio semántico, pero al contrario que los anteriores hace uso técnicas de aprendizaje máquina mediante inducción para la creación de anotaciones de metadata.

Mas información sobre este campo, y las distintas aproximaciones y sistemas que implementan estas técnicas, se puede encontrar en (Kiryakov et al., 2004; Cohen y Sarawagi, 2004; Michelson y Knoblock, 2005; Reeve y Han, 2005).

### 3.3.3. El sistema Emotag

La aplicación **Emotag**, (Francisco y Hervás, 2007), desarrollada en la Facultad de Informática de la Universidad Complutense de Madrid, es un sistema de anotación semántica de emociones a través del cual se pueden obtener las emociones inmersas en un texto mediante el uso de diferentes técnicas de Procesamiento del Lenguaje Natural. Este sistema posee, además, la capacidad de analizar y etiquetar texto tanto en inglés como en español.

**Emotag** abstrae las dos teorías principales psicológicas sobre la clasificación y reconocimiento del espacio emocional humano: la teoría de las

---

<sup>1</sup><http://wordnet.princeton.edu/>

<sup>2</sup><http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-htmls/onto.index.html>

<sup>3</sup>[www.google.es](http://www.google.es)

<sup>4</sup><http://gate.ac.uk/>

categorías emocionales y la teoría de las dimensiones emocionales, descritas ambas en la sección 3.3.1. De esta manera, en el sistema se puede elegir qué tipo de etiquetado emocional se desea realizar sobre el texto a analizar, obteniendo en cada caso resultados basados en dichas teorías. En el caso de las categorías emocionales, el programa genera como salida la emoción predominante, según el análisis realizado del texto, de un conjunto de categorías emocionales organizadas jerárquicamente e implementada mediante una ontología denominada **OntoEmotions**, la cual abarca todas las diferentes clasificaciones expuestas en esta línea. Por el contrario, en el caso de las dimensiones emocionales, el programa se basa en el estándar SAM, expuesto en la sección 3.3.1, para realizar el análisis, devolviendo en este caso una tripla con los distintos valores de evaluación, activación y control inmersos en el texto analizado.

Para poder discernir cual es la emoción predominante en un texto, en el caso de las categorías emocionales, o los valores de las tres dimensiones, en el caso de las dimensiones emocionales, el programa se vale de un *corpus* de cuentos etiquetado previamente por humanos. Este *corpus* es denominado en el sistema como *List of Emotional Words (LEW)*, y es generado mediante la siguiente sucesión de pasos.

- **Etiquetado de los cuentos:** En primer lugar, los cuentos son etiquetados por humanos, los cuales deben asignar a cada fragmento de los cuentos propuesto por la aplicación una categoría emocional, o una tupla de valores de evaluación, activación y control.
- **Obtención de la emoción más asignada en cada frase:** En el caso de las dimensiones emocionales, se calcula la media de cada frase, mientras que en las categorías emocionales se realiza el siguiente proceso: si al menos la mitad de los evaluadores han considerado una emoción predominante en el fragmento, se obtiene esa emoción. Si no, se sube un nivel en los conceptos de la ontología por cada emoción, propagando las elecciones de los evaluadores. Si así hay al menos la mitad de evaluadores que han elegido una emoción, se selecciona la de menor nivel. Este proceso continúa hasta que existe alguna emoción con al menos la mitad de evaluadores de acuerdo.
- **Inserción en la LEW:** Una vez asignadas las etiquetas y triplas de dimensiones, se analiza cada frase con la aplicación Minipar, eliminando todas aquellas palabras cuya categoría gramatical aparezca en la lista de parada del programa; es decir, todo menos los *nombres*, *verbos*, *adverbios* o *adjetivos*. Una vez analizada cada palabra, se obtiene la raíz de ésta y su categoría gramatical, las cuales serán insertadas en la lista LEW junto con la categoría asignada a la palabra, o la tripla de valores de las dimensiones.



- **Extension de la LEW con WordNet:** Una vez realizado todo este análisis, se buscan los sinónimos y antónimos de cada palabra en WordNet, asignándoles los mismos valores que a la palabra a los primeros, y los valores opuestos a los segundos.

Finalizado todo este proceso, la lista LEW almacenará, en el caso de las dimensiones emocionales, la media de la tripla de valores asignados por los evaluadores, mientras que en el caso de las categorías emocionales almacenará la probabilidad de que ese par, palabra-categoría gramatical, sea designada con esa emoción.

El proceso de asignación de una categoría emocional o una tripla de las dimensiones SAM a un texto se realiza en base a esta lista LEW descrita anteriormente, a través de los siguientes pasos:

- **Análisis con Minipar:** Cada frase es analizada con la aplicación Minipar, obteniendo de esta manera las palabras afectadas por una negación, así como la raíz y categoría gramatical de cada una en la oración. En función de estos pares, se continúa con el análisis del texto.
- **Eliminación de las palabras no influyentes:** Si alguna de las categorías gramaticales de las palabras analizadas en la oración aparecen en la lista de parada del programa, son eliminadas del análisis.
- **Obtención de la emoción del par:** Con todos los pares obtenidos en el proceso anterior, se consulta su existencia en la lista LEW. En caso afirmativo, se asigna a esa palabra las probabilidades de las distintas emociones existentes para ese par en el caso de las categorías emocionales, y los valores de evaluación, activación y control en el caso de las dimensiones emocionales.
- **Extensión con la lista ANEW:** En el caso de de las dimensiones emocionales, si la palabra buscada no aparece en la lista LEW se busca en la lista ANEW, explicada en la sección [3.3.1](#).
- **Extensión con WordNet:** Si en ambos casos el par no aparece en la lista LEW, ni en la lista ANEW en el caso de las dimensiones, se obtiene el hiperónimo de la palabra mediante WordNet y se vuelve a realizar la búsqueda en la lista LEW, y también en la lista ANEW en el caso de las dimensiones. Así, se continúa hasta que se encuentra el primer hiperónimo válido, insertando la nueva palabra en la lista LEW con los valores de éste.
- **Eliminación del par:** Si ninguno de los procesos descritos ha resultado satisfactorio, la palabra se elimina del proceso de marcado.

Una vez se han obtenidos todas las categorías y las triplas de cada palabra, el proceso continúa evaluando la frase completa. En el caso de las dimensiones emocionales, la tripla final de la frase se genera mediante la realización de la media de todos los valores de las palabras que la componen. En el caso de las categorías emocionales, todas las probabilidades de cada emoción que aparezca en cada palabra son sumadas, obteniendo así el total de cada emoción en la frase total. A continuación, se realiza un proceso de ascensión en la ontología, donde todas las probabilidades de los nodos hijos de una emoción son sumadas a su valor, continuando con este proceso hasta las emociones primarias. Para finalizar se elige la emoción con mayor probabilidad y con el nivel más bajo en la jerarquía.

Mediante este proceso **Emotag** es capaz de anotar textos con las emociones que con mayor probabilidad etiquetaría un ser humano, utilizando para ello los diferentes recursos semánticos y las distintas técnicas de Procesamiento del Lenguaje Natural descritas, y pudiendo obtener de esta manera tanto la categoría emocional más probable como la tripla de dimensiones emocionales promedio.

---

## Capítulo 4

# REC: Sistema Automático de Gestión de Cámara en Entornos Virtuales Basado en Emociones

Como se ha dicho anteriormente, en este trabajo se presenta un sistema automático de dirección de cámaras capaz de decidir, en tiempo real, cuál es la mejor configuración de cámara para una situación concreta del guión de la historia que se está narrando, basándose para ello en técnicas cinematográficas. Este trabajo está influenciado por las pocas, pero importantes, aproximaciones que han investigado sobre gestores de cámaras basados en técnicas de cine, y que han sido presentados en la sección [3.2.2](#), si bien este trabajo se centra más, en el aspecto emocional inmerso en las diferentes técnicas cinematográficas, y en cómo obtener y tratar la información necesaria, a partir del guión de la historia que se está representando, para su correcto funcionamiento.

Por regla general, este tipo de sistemas basan sus decisiones en el aspecto geométrico y espacial de las distintas técnicas de cine, intentando evitar, de esta manera, situaciones de oclusión con objetos de la escena, o tomas en la que la disposición geométrica de los elementos de la escena no permite mostrar con claridad qué es lo que está sucediendo, como conversaciones de tres personajes en las que un actor obstruya a otro, persiguiendo, además, lograr una mejor calidad de la historia representada. Sin embargo, no suelen tener en cuenta una de las principales características de estas técnicas, y que ha dotado de la gran expresividad y riqueza visual a las películas, como es el aspecto emotivo y dramático que subyace en los distintos parámetros de las técnicas.

La obtención de la información necesaria para poder determinar en cada momento la mejor configuración de cámara para una situación dada, es

otro de los principales problemas de estos sistemas. En la mayoría de los casos, esta tarea se suele abordar mediante configuraciones predeterminadas para ciertas acciones realizadas por los elementos del entorno virtual, o es generada a mano en un proceso previo de evaluación de los acontecimientos de la historia, asignando a cada acción los parámetros correspondientes. Este problema se acrecenta más aún en las pocas aproximaciones que han introducido en cierta medida el aspecto emotivo en la gestión de cámara.

Si hace tiempo era impensable que las computadoras pudieran reconocer elementos de un texto, como frases o categorías sintácticas, o determinar las emociones inmersas en un diálogo, actualmente se ha producido un gran avance en este sentido, surgiendo numerosas técnicas y sistemas de procesamiento del lenguaje que permiten dar los primeros pasos en esta dirección. Atendiendo a esto, en este trabajo se ha intentado adaptar un conjunto de estos métodos para extraer la información necesaria para determinar, centrándose como se ha dicho en el aspecto emotivo, las mejores configuraciones de cámara en un entorno virtual según la historia narrada.

Para ello, el sistema toma como entradas dos archivos, el texto del guión, en el formato específico descrito en la sección 2.2 y categorizado mediante el lenguaje de marcado HTML proporcionado por el programa de edición de guiones **Celtx**<sup>1</sup>, y la lista de los nombres de los personajes que interactúan en ella. Estos dos archivos son procesados y analizados generando una estructura que representa los distintos elementos de los que se compone el guión de una película, determinando así las escenas, acciones, diálogos y paréntesis, los cuales son procesados y enriquecidos con etiquetas emocionales a través de las que se obtiene la intensidad emotiva de cada elemento.

A partir de esto, la aplicación crea una nueva estructura mediante la que se representa el hilo de la historia, que viene dictada por el guión, y en la que cada escena está formada por un conjunto de secuencias que representan aquellas acciones de la historia que poseen características similares a la hora de visualizarse, como por ejemplo, diálogos, escenas de movimientos, escenas estáticas, etc. Mediante esta estructura y los distintos eventos recogidos por el sistema en la reproducción virtual en 3D de la historia, la aplicación es capaz de decidir en tiempo real cuál es la mejor forma de filmar las acciones que están sucediendo en la escena, proporcionando, en cada caso, una configuración de cámara formada por un conjunto de parámetros que abstraen el conocimiento de las diferentes técnicas cinematográficas, y que será interpretada por el gestor de cámaras del entorno virtual implementado.

Como se puede ver en la Figura 4.1, el sistema se ha diseñado e implementado en dos módulos principales: el módulo lógico y el módulo gráfico. Todos los procesos de identificación de los elementos del guión, análisis de éstos, sincronización del flujo de la historia y toma de decisiones, son llevados a cabo en el primero de ellos, mientras que todos los procesos que

---

<sup>1</sup><http://www.celtx.com/>

involucran la interacción con el motor gráfico son realizados en el segundo.

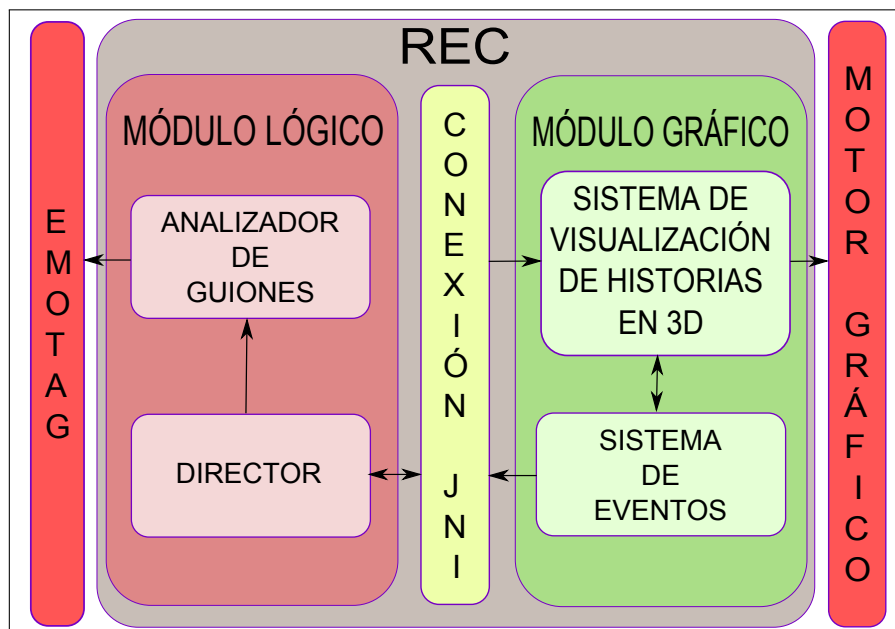


Figura 4.1: Arquitectura del sistema

El módulo lógico está formado por dos componentes principales, el *Analizador de Guiones* y el *Director*, en los que se dividen las dos etapas principales de este módulo. En una primera fase, el *Analizador de Guiones* realiza un proceso de análisis del texto del guión, como se ha dicho anteriormente, generando a través de él una estructura lógica que permite representar los diferentes elementos que lo componen, extrayendo cierta información relevante de cada uno, como por ejemplo, los actores que lo componen o el texto del elemento. En una segunda fase, esta estructura es enriquecida mediante un conjunto de etiquetas emocionales asignadas a cada texto de cada componente utilizando el programa **Emotag**, y calculando a partir de éstas la intensidad emocional inmersa en el texto del guión.

El *Director* es el componente que transforma esa estructura de elementos de un guión en una historia real, compuesta por escenas y secuencias, y, a partir de la cuales, junto con la información proporcionada por los eventos del módulo gráfico, determina en tiempo real la mejor manera de filmar la secuencia, devolviendo una configuración con los parámetros necesarios para gestionar la cámara virtual.

El módulo gráfico, como puede observarse en la Figura 4.1, está compuesto principalmente por el *Sistema de Visualización de Historias en 3D*, que permite reproducir el hilo de una historia mediante la carga de los objetos y personajes que la componen, así como las diferentes acciones de cada uno de ellos en el tiempo. El sistema de gestión de cámaras en este sistema ha sido

adaptado para responder a las peticiones del módulo lógico, permitiendo obtener toda la riqueza de las técnicas de cine implementadas. Además, posee un *Sistema de Eventos* que captura las diferentes acciones lanzadas por cada personaje, u objeto, e informa a la parte lógica del comienzo y finalización de su ejecución, a partir de los cuales se produce la sincronización entre los dos módulos.

Debido a la necesidad de utilizar un motor gráfico para la visualización de la historia (para poder así obtener toda la funcionalidad del sistema de gestión de cámaras virtuales), y a que la gran mayoría de los motores gráficos están implementados en C++, el sistema ha sido desarrollado en los lenguajes de programación Java y C++, conectando ambos módulos mediante Java Native Interface (JNI).

Para poder probar la viabilidad del sistema, y su correcto funcionamiento, se ha simulado, mediante el módulo gráfico, un fragmento de la película “El Señor de los Anillos: El retorno del rey”, contrastando los resultados generados con un caso real de una película con un amplio abanico de técnicas cinematográficas en su gestión de cámaras. La interpretación de estos resultados, así como su discusión, se detallarán en el Capítulo 5, donde se comparará la simulación realizada con la subjetividad humana de un director real.

Este capítulo se divide en tres secciones donde se explicarán los diferentes módulos que componen el sistema, así como el funcionamiento de éstos. En las dos primeras secciones se describirán en detalle los diferentes componentes de estos módulos, presentando su arquitectura y explicando su funcionamiento. En la última sección se mostrarán las diferentes fases en las que se divide el algoritmo de gestión de cámara basada en emociones, a través del cual se obtienen las diferentes configuraciones de cámara acordes a la secuencia, así como su visualización en la parte gráfica del sistema.

## 4.1. Módulo lógico

Como ya se ha introducido anteriormente, en este módulo se llevan a cabo todos los procesos de análisis del guión, creación de la estructura lógica de éste, etiquetado emocional, creación de la estructura lógica de la historia y elección, en tiempo real, de la configuración de cámara más adecuada a lo que está sucediendo en la escena. Para ello, este módulo se ha dividido en dos componentes claramente diferenciados, según su tarea y su momento de ejecución, el *Analizador de Guiones* y el *Director*. A lo largo de esta sección se detallarán las distintas partes que forman cada componente, así como sus conexiones con los distintos elementos utilizados, dando una perspectiva general de la arquitectura de cada uno.

Se ha intentado desde un principio que este módulo fuera lo más genérico posible, proporcionando así la posibilidad de ser utilizado por cualquier com-

ponente gráfico que permita visualizar una historia, con una mera adaptación de su gestor de cámara y la creación de un sistema de eventos. Por esto, se ha abstraído toda la información relevante a las técnicas cinematográficas en una estructura de datos denominada *Configuración de Cámara*, en la que se integran los parámetros más importantes de éstas, como por ejemplo, los tipos de movimientos, la distancia a los objetos, los objetos que componen la secuencia, etc., y que permite, con una pequeña adaptación en cada caso, una fácil interpretación, desde el punto de vista geométrico, por parte de los gestores de cámaras de los entornos virtuales. La adaptación de esta estructura de datos, pese a parecer todo lo contrario, no supone una complicación, al proporcionar casi todos los entornos gráficos funcionalidades de más alto nivel en la gestión de cámara como es el *tracking* sobre objetos del entorno, la posibilidad de asignar la orientación de la cámara hacia un objeto, el posicionamiento de ésta en el espacio mediante un sistema de coordenadas, etc.

De esta manera, este módulo realiza la selección en tiempo real de la mejor configuración de cámara, relegando el aspecto geométrico al adaptador de cámara del entorno virtual, y centrándose, por el contrario, en aspectos de más alto nivel de las técnicas cinematográficas como el número de personajes, la intensidad emocional, el tipo de secuencia, etc. Para ello, realiza un análisis de los dos ficheros de entrada, el guión y la lista de actores de la historia, junto con los eventos recibidos del sistema gráfico en tiempo real para la correcta sincronización, produciendo en cada caso una estructura de datos con la configuración de cámara según los diferentes parámetros de las técnicas de cine.

#### 4.1.1. Analizador de Guiones

Existen muchas formas de presentar una historia en texto escrito, desde un cuento, una novela, una obra de teatro o un guión, pero quizá, de todas ellas, ésta última sea la que mejor estructurada presenta la información del hilo y los acontecimientos de la historia, y su motivación radica en el hecho de que debe ser revisado e interpretado por un grupo muy amplio de personas que conforman el conjunto del rodaje de una película. Debido a esto, progresivamente el guión ha adquirido un formato estándar (ver sección 2.2) que permite una fácil interpretación, tanto de los componentes de una escena como de las acciones de los mismos, y que ha fomentado el desarrollo de diferentes aplicaciones que facilitan la creación y gestión de guiones, permitiéndolos exportar a formatos más manejables por los ordenadores.

En este trabajo se pretende dar una primera aproximación de las diferentes posibilidades que proporciona este formato, a partir del cual se puede extraer mucha información relevante para la gestión de las configuraciones de cámara, y que permite dotar de una mayor calidad a la filmación. Para ello, se ha desarrollado en la aplicación este componente, mediante el cual

se genera una estructura lógica de un guión con los diferentes elementos que lo componen: las escenas, las acciones, los diálogos y los paréntesis. A su vez, en este proceso de análisis se extrae la información más relevante para la gestión de la cámara, como los actores que componen cada elemento, el texto de los diálogos, las descripciones de las acciones o la intensidad emocional de cada elemento, que posteriormente será utilizada en el proceso de selección de la configuración de cámara más adecuada.

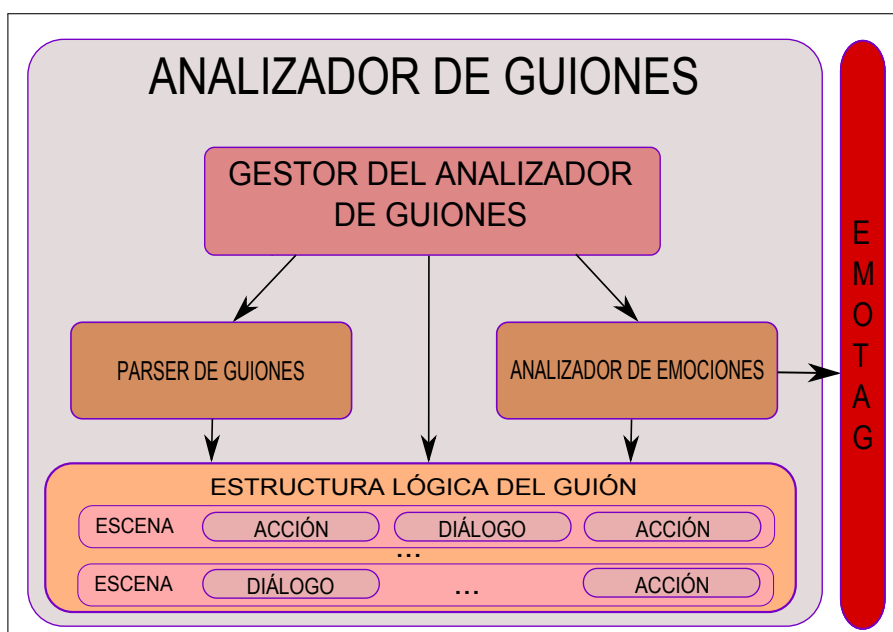


Figura 4.2: Elementos de *Analizador de Guiones*.

Como se puede ver en la Figura 4.2, este componente está formado por dos elementos principales, el *Parser de Guiones* y el *Analizador de Emociones*, cuya interacción con la estructura lógica del guión está guiada por el *Gestor del Analizador de Guiones*. Además, en este componente se realiza toda la interacción con el programa **Emotag**, a través del elemento *Analizador de Emociones*, a partir del cual se podrá evaluar la intensidad emocional inmersa en los textos de cada elemento del guión.

De esta manera, mediante el *Gestor del Analizador de Guiones*, que guiará el proceso de generación de la estructura lógica del guión, en este componente se realiza todo el análisis y procesamiento de la información relevante contenida en el guión. Seguidamente, se generará la estructura lógica de la historia que permitirá identificar lo que está sucediendo en tiempo real y seleccionar la mejor configuración de cámara.

Para ello, el *Parser de Guiones* realiza un primer análisis del texto del guión en formato HTML, identificando sus elementos, así como sus características relevantes, y generando la estructura lógica del guión, la cual



será utilizada por el *Analizador de Emociones* para analizar los textos de cada elemento, extrayendo las emociones básicas inmersas mediante **Emo-tag**, y generando la intensidad emocional de cada uno. Una visión más en profundidad de todo este proceso se mostrará en las secciones 4.3.1 y 4.3.2, donde se exponen las dos primeras etapas del algoritmo de gestión de cámaras implementado en el sistema.

#### 4.1.2. Director

Al igual que en el proceso de creación de una película existe el papel de director, cuya función consiste en evaluar qué es lo que está sucediendo, cómo debe suceder y cómo debe filmarse para ser visualizado de la mejor manera posible, en este sistema se ha diseñado el componente *Director* encargado de centralizar todo el proceso de creación de la estructura lógica de la historia, la sincronización con el entorno virtual a partir de los eventos y la selección de la mejor configuración de cámara de acuerdo a la secuencia que está sucediendo en cada momento.

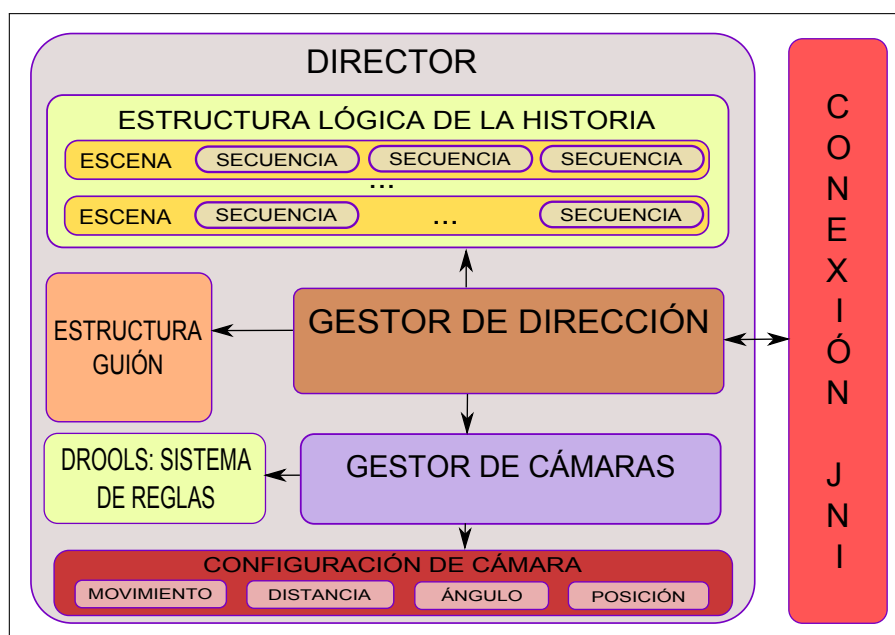


Figura 4.3: Elementos del *Director*.

Para poder realizar todas estas tareas, este componente se ha dividido en dos elementos principales, como se puede ver en la Figura 4.3, según las diferentes acciones realizadas en el proceso. El elemento *Gestor de Dirección* es el encargado de centralizar todas estas acciones, interactuando para ello con el elemento *Gestor de Cámaras*, las estructuras de datos del guión y la historia, y con el módulo gráfico del sistema. De esta manera, este elemento

realiza las tareas de creación de la estructura lógica de la historia a partir de la estructura lógica del guión creada en el *Analizador de Guiones*, recepción de los eventos del entorno virtual mediante la conexión JNI, sincronización en tiempo real de la estructura de la historia con el entorno virtual a partir de estos eventos, y comprobación de la necesidad de actualizar la configuración de cámara o no.

Si esta última comprobación produce un resultado verdadero, el *Gestor de Dirección* proporciona la información de la secuencia actual al elemento *Gestor de Cámaras*, cuya función es en determinar cuál es la mejor manera de visualizar la escena de la historia que se está narrando, generando, si fuera necesario, una nueva configuración de cámara en cada caso. Para ello, se ha utilizado el sistema de gestión de reglas de código abierto **Drools**<sup>2</sup>, el cual proporciona una fácil interacción con el lenguaje de programación Java, en el que está implementado este módulo, y una rapidez similar a otros sistemas de reglas conocidos, como **Jess**<sup>3</sup>, al utilizar una versión optimizada del algoritmo RETE. Además de estos beneficios, se ha optado por este sistema de reglas por la facilidad de su sintaxis, el cual, incluso, permite definir lenguajes específicos de dominio (DSL), pudiendo de esta manera generar reglas en un lenguaje similar al utilizado por los humanos, aunque esta funcionalidad no ha sido utilizada en el sistema.

```
rule "Action_Static_MEDIUM_CLOSE_SHOT"
    salience 25
    when
        a:ActionSeq{(numActors(>=1)) && {(emotionalIntensity>=80)||
        (emotionalIntensity<=-80)}};
        CameraConfiguration(actualRule matches "Action_Static\S*");
    then
        CameraConfiguration cameraConfiguration = new CameraConfiguration();
        cameraConfiguration.setDistance(new Distance(Shots.MEDIUM_CLOSE_SHOT));
        cameraConfiguration.setTime(7000);
        cameraConfiguration.setAngle(0);
        cameraConfiguration.setPrincipalObjects(a.getActors());
        cameraConfiguration.setLookAtShot(0);
        cameraConfiguration.setActualRule("Action_Static_MEDIUM_CLOSE_SHOT");
        listCamConf.add(cameraConfiguration);
        System.out.println("Action_Static_MEDIUM_CLOSE_SHOT");
    end
```

Figura 4.4: Ejemplo de la sintaxis de reglas del sistema Drools para este sistema.

Un ejemplo de la sintaxis del sistema de reglas utilizado se puede observar en la Figura 4.4, y en mayor número en el Apéndice A, donde se aprecian los elementos que componen las reglas. Éstas se dividen en dos partes diferenciadas: por un lado, las premisas o condiciones, que son características

<sup>2</sup><http://www.jboss.org/drools/>

<sup>3</sup><http://herzberg.ca.sandia.gov/>

que tienen que cumplir los hechos, que en Drools son *Beans* de Java, para disparar la regla; y por otro lado, las consecuencias, que son las acciones que se desean realizar en caso de que dicha regla sea activada. Las premisas están precedidas por la palabra reservada *when*, mientras que las conclusiones están precedidas por la palabra *then*. Además, existen diversos atributos asociados a las reglas que permiten gestionar grupos de estas, prioridades, etc. En este caso, se ha utilizado el atributo *salience*, que permite asociar un número indicador de la relevancia y orden de evaluación de la regla, puesto que en muchas ocasiones se darán circunstancias que puedan cumplir varias premisas a la vez, al ser algunas más específicas que otras. En estos casos, se dará prioridad a estas últimas, ya que generan una mejor visualización de lo que está sucediendo, al cumplir mayor número de condiciones.

Se ha elegido un sistema de reglas para este componente del sistema debido a su facilidad de uso e interacción, a la perfecta adecuación del problema con la resolución de éste mediante reglas, y a la capacidad que proporcionan este tipo de sistemas para implementar nuevas reglas con un simple conocimiento de los elementos utilizados para su desarrollo. En el caso de las secuencias, que son los elementos evaluados en las condiciones, y de las configuraciones de cámara, que son los objetos generados por las reglas, no es necesario tener mayor conocimiento del sistema, pudiendo de esta manera modificar, o ampliar, las reglas con un sencillo proceso.

Todas estas acciones realizadas por el componente *Director*, así como su funcionalidad dentro del proceso global del sistema, se detallarán más en profundidad en las secciones 4.3.3 y 4.3.4, donde se describen las etapas tres y cuatro del algoritmo de gestión de cámaras implementado en el sistema.

## 4.2. Módulo gráfico

Como se ha dicho anteriormente, el módulo gráfico está formado por dos componentes principales, el *Sistema de Visualización de Historias en 3D* y el *Sistema de Eventos*, a través de los cuales se puede reproducir una historia en un entorno virtual en 3D, así como gestionar los diferentes eventos lanzados en la reproducción y representar las distintas configuraciones de cámara propuestas por el módulo lógico. Para poder realizar todas estas tareas, y obtener la funcionalidad necesaria para generar las distintas técnicas de cine implícitas en las reglas, esta aplicación se ha diseñado e implementado sobre el motor gráfico de libre distribución **Ogre**<sup>4</sup>.

*Ogre* es uno de los motores gráficos de código abierto más potentes y versátiles que existen actualmente. Si bien es cierto que un prototipo del sistema con un motor comercial mejoraría en rendimiento y velocidad, se ha optado por este motor tanto por sus beneficios económicos como por su potencia y facilidad de uso. Además, *Ogre* posee una amplia comunidad de

---

<sup>4</sup><http://www.ogre3d.org/>

apoyo y generación de extensiones para el motor, junto con gran cantidad de documentación que hace realmente fácil su uso y comprensión.

Un pequeño esquema de los elementos más importantes de este módulo y sus respectivas conexiones se muestra en la Figura 4.5. Como se puede observar, los dos componentes principales de este módulo, el *Sistema de Visualización de Historias en 3D* y el *Sistema de Eventos*, están conectados con el módulo lógico mediante el sistema JNI, los cuales, a su vez, están interconectados para poder determinar y gestionar los diferentes eventos que se producen en el entorno virtual, y que serán relevantes en la sincronización de la estructura de la historia del módulo lógico con la reproducción en tiempo real del módulo gráfico.

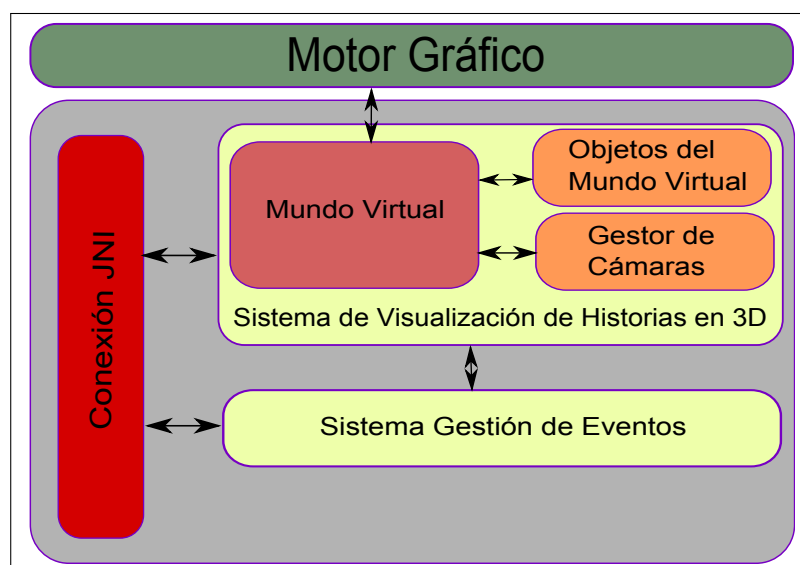


Figura 4.5: Esquema Módulo Gráfico

Atendiendo a esto, se ha diseñado e implementado un sistema capaz de reproducir una historia en un entorno virtual a través de la carga de los modelos en 3D de ésta y de las distintas acciones de cada personaje en el tiempo. Para ello, se ha establecido un conjunto de objetos, con unas características determinadas, dependiendo de las distintas acciones que cada personaje, u objeto, pueda realizar, así como un conjunto de acciones permitidas en el sistema, y que se expondrán en mayor detalle en las siguientes secciones. Además, se ha desarrollado un gestor de cámaras capaz de interpretar y generar las distintas configuraciones de cámara propuestas por el módulo lógico, junto con un sistema de eventos que permite informar a este último de lo que está sucediendo en la escena en cada instante.

Esta parte del sistema se ha intentando que sea lo más funcional posible, permitiendo evaluar los diferentes beneficios e inconvenientes del trabajo en una simulación real sobre un entorno virtual adaptado para el algoritmo de

gestión de cámaras propuesto en esta memoria, relegando todos los procesos de toma de decisión a la parte lógica, y asociando a ésta un carácter meramente ejecutivo desde el punto de vista de la gestión de cámaras.

#### 4.2.1. Sistema de Visualización de Historias en 3D

Ésta es, quizá, la parte más importante del módulo gráfico, ya que proporciona la información necesaria para la sincronización de eventos y permite representar las decisiones tomadas en la parte lógica, además de visualizar una historia configurada previamente mediante la carga de los modelos en 3D y sus acciones en el tiempo.

Como se puede ver en el esquema de la Figura 4.5, el componente está formado por dos elementos principales. El primero de ellos, el *Mundo Virtual*, gestiona toda la interacción con el motor gráfico y recibe las peticiones de refresco de cada iteración del módulo lógico. En cada una de estas iteraciones se produce un análisis de todos los objetos que conforman el mundo virtual, actualizando todos aquellos objetos o personajes que lo precisen, así como el estado de las acciones de cada uno, e informando de ello al *Sistema de Eventos*. El *Mundo Virtual* también recibe las configuraciones de cámara proporcionadas por el módulo lógico, actualizando la información del *Gestor de Cámaras* del componente, que a su vez interpreta la nueva configuración de cámara y realiza las acciones precisas para representarla.

A lo largo de esta sección se describirá en detalle el funcionamiento del *Sistema de Visualización de Historias en 3D*, exponiendo en cada caso los aspectos más importantes de su funcionamiento y diseño. De esta manera, se comenzará explicando los diferentes tipos de datos que el sistema acepta como entrada y la forma en que se cargan en la aplicación. Se continuará en el siguiente punto exponiendo los distintos tipos de elementos del componente, según su función y modo de ejecución, así como los distintos tipos de acciones que puede realizar cada elemento y cómo se deben asociar a cada objeto o personaje. Finalmente, se describirán las funcionalidades más importantes del *Gestor de Cámaras*.

#### Carga de los elementos de una historia

Como cualquier otro sistema virtual en 3D dimensiones, los datos fundamentales del sistema son los modelos en 3D de los objetos y personajes, que serán cargados en el motor gráfico para su posterior visualización. De esta manera, este componente necesita como entrada los modelos en 3D de la historia que se va a representar, así como sus acciones en el tiempo que permitan representar el flujo de la historia. El motor gráfico utilizado posee un formato propio de representación de modelos que permite un amplio número de posibilidades, como por ejemplo, animaciones por esqueleto o por vértices, transparencias en las texturas, coordenadas originales del entorno

de modelado o iluminación en la textura, entre otras muchas.

Para poder realizar la carga de los objetos implicados en el sistema, es necesaria una transformación previa al formato del motor gráfico. Existen multitud de extensiones gratuitas, y descargables desde la página del motor, que permiten transformar los formatos de los distintos entornos de modelado al formato de *Ogre*.

La carga de los diferentes objetos y personajes en la aplicación, así como sus posiciones iniciales, direcciones, rotaciones o tamaños, se realiza manualmente mediante código. Si bien es cierto que sería interesante un sistema de carga más potente que utilizara, por ejemplo, archivos de configuración en XML, este tipo de mejoras se deja para un trabajo futuro, ya que no son determinantes de cara a evaluar la viabilidad del sistema. Los diferentes modelos cargados se pueden configurar con una posición inicial (en el sistema de coordenadas del motor), una dirección inicial (aquellos que pueden realizar movimientos), una rotación y una escala para el tamaño. Si alguno de estos parámetros no se inicializa, el componente por defecto utiliza las variables creadas en el sistema de modelado. Esto permite que se importe directamente un mundo virtual generado completamente en el entorno de modelado, sin necesidad de posicionar y escalar todos los objetos, de nuevo, en el proceso de carga.

Para facilitar y poder estructurar este proceso se ha implementado el paquete *LoadData* en el que sólo hay una clase denominada, "LoadData", que debe implementar el método *createScene(SceneManager\* mSceneMgr, EventSystem \*eventSystem, SoundManager\* soundMgr)*.

### Tipos de elementos

Actualmente, los motores gráficos incorporan multitud de funciones que implementan las diferentes técnicas de representación de modelos en 3D, y que permiten realizar todas las posibilidades que éstos proporcionan, obteniendo así mayor realismo y rendimiento. Para ello, se suelen clasificar los diferentes objetos dependiendo de las opciones que pueden realizar, como por ejemplo animaciones, rotaciones, partículas, etc., y de su forma de ejecución. Para el cometido de esta aplicación, se ha realizado la siguiente clasificación de acciones relevantes según su similaridad de ejecución en el motor, y sobre las que se basan los diferentes tipos de objetos del componente.

- **Acciones de movimiento:** En este grupo se clasifican todos los objetos que tienen la capacidad de trasladarse o rotar en el entorno virtual.
- **Acciones de animación:** Este grupo está formado por todos aquellos objetos que pueden realizar animaciones, como por ejemplo, hablar, sentarse, luchar, etc., en el entorno virtual.

Con esto, se ha creado para este componente una jerarquía de clases que va especializando los objetos según sus funcionalidades (permitiendo

un amplio rango de opciones a la hora de adoptar la representación de un elemento(. Como se muestra en el diagrama de clases de la Figura 4.6, el elemento principal del sistema es el objeto “GeHi3DObject”, del que heredan el resto, y que abstrae todas las variables necesarias para la carga y refresco de un modelo en el entorno virtual. Esta clase posee los métodos abstractos de *load(SceneManager\* sManager)* y *update(SceneManager\* sManager, floattime, constFrameEventvt)* que deben ser implementados en todas las clases que la hereden, particularizando en cada caso según sus propiedades y funciones específicas.

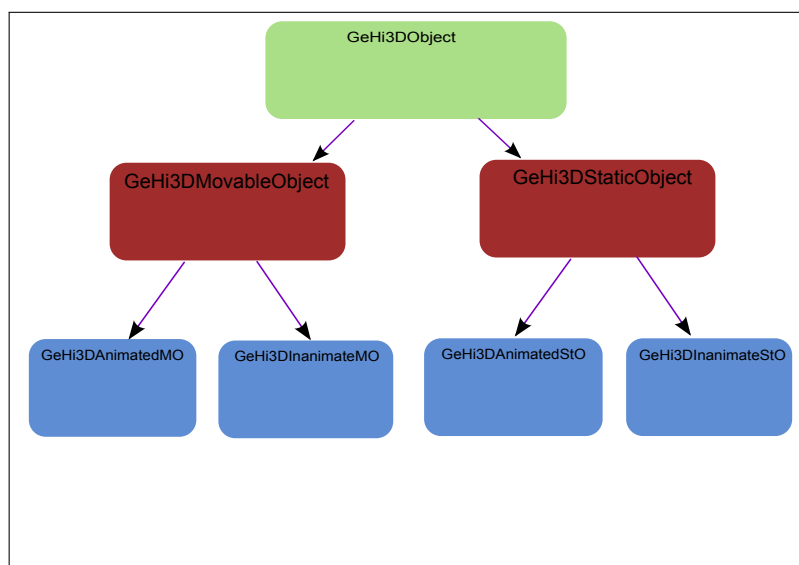


Figura 4.6: Jerarquía de clases de los diferentes objetos de la aplicación

Como se puede ver en la Figura 4.6, se ha optado por una estructura jerárquica que implementa las distintas opciones de elementos según la clasificación expuesta anteriormente y evitando la herencia múltiple, que aunque permitida en C++, puede llevar a ciclos en la herencia y suele ser propensa a errores. Así, los distintos tipos de objetos posibles en la aplicación son:

- **Objetos movibles:** En este grupo se engloban todos aquellos objetos o personajes que pueden realizar acción de movimiento o rotación. La clase “GeHi3DMovableObject” abstrae este comportamiento y es extendida por las dos clases siguientes.
  - **Objetos movibles animados:** Estos objetos tienen la funcionalidad de poder realizar acciones de movimiento y animación, independientemente o a la vez. Este comportamiento ha sido implementado en la clase “GeHi3DAnimatedMO”.
  - **Objetos movibles inanimados:** Estos objetos sólo implementan la funcionalidad de movimiento, no pudiendo tener acciones

que impliquen animaciones. Este comportamiento ha sido implementado en la clase “GeHi3DInanimateMO”.

- **Objetos estáticos:** En este grupo se encuentran todos los objetos estáticos; es decir, que no pueden realizar ninguna acción de movimiento como transacción o rotación. La clase “GeHi3DStaticObject” abstrae este comportamiento y es extendida por las dos clases siguientes.
  - **Objetos estáticos animados:** Este tipo de objetos sólo permite aquellas acciones que impliquen animaciones, sin ningún tipo de movimiento. Este comportamiento ha sido implementado en la clase “GeHi3DAnimatedStO”.
  - **Objetos estáticos inanimados:** Aunque parece contradictorio tener un objeto que no posea ninguna funcionalidad, estos objetos son necesarios en este tipo de entornos y permiten hacer una distinción para acelerar el proceso de refresco al poder descartarlos automáticamente. Este comportamiento ha sido implementado en la clase “GeHi3DInanimateStO”.

Este tipo de clasificaciones jerárquicas tienen la ventaja de automatizar la discriminación de objetos a la hora de refrescar el mundo virtual, lo que se traduce en una mejora del rendimiento y la rapidez del sistema. Es por ello que se ha optado por esta estructura de clases para determinar los diferentes tipos de objetos en este componente.

### **Tipos de acciones**

Hasta aquí se ha expuesto un sistema que simplemente es capaz de cargar y visualizar elementos en un entorno virtual. Sin embargo, el propósito de este componente consiste en poder representar una historia dinámica a lo largo del tiempo. Para lograrlo se han diseñado un conjunto de acciones, que pueden realizar los diferentes objetos de la aplicación según su tipo, que van asociadas a los objetos o personajes y cuya ejecución está sincronizada con la historia mediante el tiempo. Estas acciones se pueden asociar a los objetos permitiendo programar las diferentes situaciones de una historia, y proporcionando el dinamismo buscado. Para ello, es necesario evaluar en cada iteración de refresco del módulo gráfico, aquellos objetos que posean esta funcionalidad, y comprobar si se debe comenzar la ejecución de alguna acción o no.

Atendiendo a la clasificación expuesta en la sección anterior, se han creado un conjunto de acciones que permiten dotar de esa funcionalidad a los distintos tipos de objetos permitidos en la aplicación. Las diferentes acciones implementadas en el sistema son:



- **Acción de traslación:** Asignando a la acción un vector con la nueva posición del objeto, o personaje, el sistema desplazará, con una velocidad predeterminada, el objeto a lo largo de una línea recta.
- **Acción de rotación:** En este caso, el vector asignado a esta acción se utiliza para calcular el ángulo entre el vector de dirección del objeto, o personaje, y el nuevo punto, girando éste dicho ángulo.
- **Acción de animación:** En este caso, los parámetros necesarios son el nombre de la animación en el modelo y un booleano indicando si ésta se ejecutará en un bucle o no, produciendo en el instante de tiempo indicado la ejecución de dicha animación.

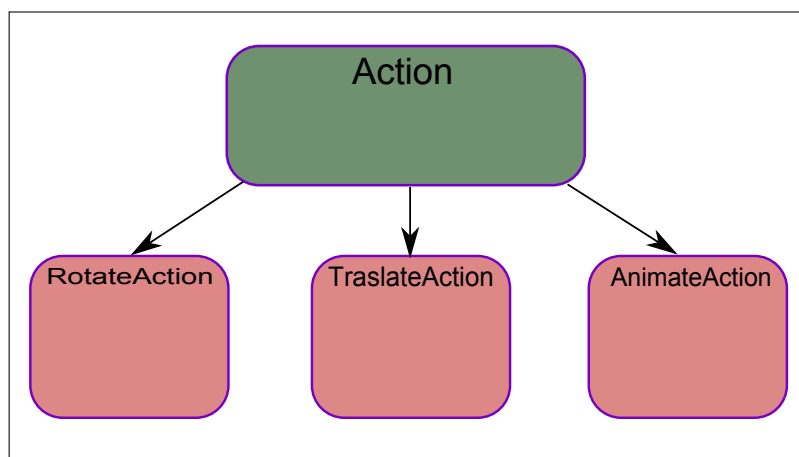


Figura 4.7: Jerarquía de clases de las diferentes acciones de la aplicación

En la Figura 4.7 se puede ver la jerarquía de clases del sistema de acciones, en la que el objeto principal es la clase “Action”, que incluye los atributos necesarios para la gestión de las acciones, así como las funciones abstractas que deberán implementar las clases que hereden de ella. Las clases “AnimationAction”, “TranslateAction” y “RotationAction”, que heredan de la clase “Action”, implementan los métodos necesarios para gestionar las acciones de la clasificación anterior, especializando en cada caso sus atributos dependiendo de su funcionalidad.

El conjunto de acciones que cada elemento realiza se almacena en una tabla hash en el propio objeto, en la que la clave utilizada es el tiempo y el objeto, la acción a ejecutar. Así, en cada iteración se consultará, en aquellos objetos que puedan realizar acciones, la tabla hash comprobando si es preciso o no realizar alguna acción. El proceso de carga de las acciones se realiza de la misma manera que en los objetos, en la clase “LoadData” del paquete del mismo nombre, mediante el método `setActions(map < int, vector < Action* > * > acts)` de los objetos del sistema.

### Gestor de Cámaras

Para poder evaluar correctamente la viabilidad del sistema, el *Gestor de Cámaras* del *Sistema de Visualización de Historias en 3D* debe ser capaz de responder a las peticiones de configuraciones de cámara propuestas por el módulo lógico. Por ello, en cada iteración el *Gestor de Cámaras* debe comprobar si hay o no una nueva configuración de cámara, interpretándola y ejecutándola, en caso afirmativo, de acuerdo a los parámetros de las reglas cinematográficas establecidas en la configuración, y actualizando la ya existente en caso de no encontrar una nueva.

Debido a esto y a que la comunicación en este sentido, al igual que en sentido contrario como se mostrará más adelante, se realiza mediante el lenguaje de marcado XML, en este componente se ha implementado un pequeño y sencillo *parser* de la estructura de datos de las configuraciones de cámara, capaz de identificar los diferentes elementos contenidos en ésta para su posterior ejecución.

A su vez, se ha adaptado, con las diferentes funciones proporcionadas por el motor gráfico utilizado y junto con diversas operaciones de geometría implementadas para este componente, el sistema de gestión de cámaras virtuales para que pueda realizar las diferentes acciones específicas de las técnicas cinematográficas que están recogidas en los elementos de las configuraciones de cámara del módulo lógico, como por ejemplo, los movimientos de *Pan* o *Zoom*, la elevación de la cámara respecto del eje vertical según el ángulo indicado, el posicionamiento de la cámara a la distancia del tipo de plano deseado, etc. Todas estas acciones son interpretadas en cada caso para los diferentes actores involucrados en la secuencia, y que vienen determinados en la configuración de cámara proporcionada. Por ello, el *Gestor de Cámaras* debe actualizar en cada iteración la posición y rotación de la cámara de acuerdo a la configuración establecida y a los diferentes personajes involucrados en ella, manteniendo las restricciones impuestas.

Una visión más en profundidad de las diferentes acciones y procesos realizados por este componente del módulo gráfico se mostrará en la sección 4.3.5, donde se expondrán las diferentes técnicas utilizadas para la gestión de la cámara del entorno virtual, según los parámetros de las posibles configuraciones de cámara.

#### 4.2.2. Sistema de Eventos

Como ya se ha expuesto, para poder sincronizar en tiempo real la estructura de la historia con el entorno virtual es necesario un sistema de eventos que identifique e interprete las diferentes acciones realizadas por los personajes, informando de ello al módulo lógico, para que éste pueda determinar exactamente qué es lo que está sucediendo en la historia y seleccionar la configuración de cámara más acorde a ello.

Para ello, se ha desarrollado este componente del módulo gráfico, el cual analiza en cada iteración el estado de las acciones de los personajes del *Sistema de Visualización de Historias en 3D*, a través de un simple mapeo entre éstas y los diferentes eventos diseñados, identificando y almacenando, para su posterior envío al módulo lógico a través de la conexión JNI, los distintos eventos lanzados en cada instante. Además del tipo de evento producido, para una correcta sincronización es necesario identificar el personaje que lo ha producido, por lo que los eventos se componen de dos elementos: el tipo de evento y el actor que lo ha producido, cuyo reconocimiento es sencillo, ya que es su objeto el que lo ha producido, obteniéndose de esta manera el nombre del personaje u objeto.

Para este trabajo se han especificado tres tipos de eventos, de acuerdo con la información necesaria para la sincronización e identificación de lo que está sucediendo en cada instante por parte del módulo lógico en la estructura de la historia.

- **Eventos de inicialización y finalización de diálogo:** Este tipo de eventos, como su propio nombre indica, permiten identificar cuándo un actor ha comenzado o finalizado un diálogo, entendiendo en este caso el diálogo como el monólogo del actor o personaje. Para poder identificar este tipo de eventos, el sistema realiza un mapeo de las distintas animaciones lanzadas, identificando aquellas cuyos nombres estén asociados a acciones de habla.
- **Eventos de inicialización y finalización de movimiento:** Estos eventos permiten especializar las diferentes secuencias generadas en el módulo lógico en situaciones con movimiento o estáticas. La identificación de este tipo de eventos es aun más sencilla que en el caso anterior, gracias al sistema de acciones desarrollado en este módulo, ya que se puede reconocer fácilmente cuándo un personaje va a comenzar o finalizar una acción de traslación o rotación en el proceso de refresco.
- **Eventos de cambio de localización:** Los eventos de cambio de localización permiten al módulo lógico la identificación de los personajes que entran y salen de la escena, logrando sincronizar la estructura de la historia. Por ejemplo, si un personaje entra en la escena, se producirá un cambio de localización de éste, lo que hará que el módulo lógico identifique si el actor está en la secuencia o no. Si no estuviera, se debería producir un cambio a la siguiente secuencia, la cual representará la entrada de este actor en la escena. Todo este proceso será explicado con mayor detalle en la sección 4.3.4. Este tipo de eventos son generados cuando un objeto del entorno con posibilidad de movimiento identifica que el plano sobre el que está es diferente al que se encontraba en el refresco anterior. Esto es reconocido debido a que los personajes en movimiento lanzan en cada iteración un rayo en

dirección al suelo, comprobando en cada caso el nombre del plano con el que colisiona el rayo lanzado.

Como ya se ha mencionado la comunicación de estos eventos producidos en el entorno virtual por parte del módulo gráfico al módulo lógico se realiza mediante el lenguaje de marcado XML y a través de la conexión JNI. Una explicación más detallada de esta elección, así como del papel global de este componente en el algoritmo de gestión de cámaras se expondrá en la sección [4.3.4](#).

### 4.3. Algoritmo de gestión de cámaras basado en emociones

Hasta este punto se han descrito, a lo largo del capítulo, los diferentes componentes del sistema, así como su función dentro de la aplicación, presentando en detalle cómo se han implementado y las decisiones que han llevado a su elección. En esta sección se va a presentar el algoritmo, y las distintas fases de éste, mediante el cual la aplicación es capaz de generar la configuración de cámara más adecuada para la acción que se está desarrollando en la historia.

Para poder realizar este proceso, la aplicación debe ser capaz de discernir en cada momento qué es lo que está sucediendo en la historia, y cuáles son las características de ese acontecimiento, como por ejemplo, los actores implicados o la carga emocional. Para ello, el sistema se vale de una estructura lógica que representa una historia en orden cronológico, formada por las distintas escenas que la componen, las cuales a su vez, agrupan un conjunto de secuencias, ordenadas también cronológicamente, que encapsulan los parámetros necesarios para determinar la configuración de cámara correcta.

De esta manera, las secuencias han sido definidas en este sistema como la unidad principal, y a partir de ellas se pueden seleccionar las distintas configuraciones de cámaras en la aplicación. Atendiendo a esto, y a la clasificación propuesta por Arijon en ([Arijon, 1991 \(originally published 1976\)](#)), en este sistema se han establecido dos tipos de secuencias básicas: secuencias de acción y secuencias de diálogo, que, a su vez, se pueden especializar en: acciones con movimiento, acciones estáticas, diálogo con movimiento y diálogo estático. Las primeras encapsulan todas aquellas acciones realizadas en la historia que no conllevan un diálogo entre los personajes y que se visualizan con un conjunto de técnicas determinadas, diferenciando entre las técnicas para movimiento o estáticas. Por el contrario, las segundas representan exactamente lo inverso, acciones en las que interviene un diálogo entre dos o más actores, distinguiendo al igual que en las acciones, entre las técnicas para diálogos estáticos y en movimiento.

Para poder clasificar cada una de las acciones de la historia en los dos

tipos de secuencias anteriores, el algoritmo realiza en la primera fase un análisis del guión de la historia, determinando para cada escena los distintos elementos que la componen, como los diálogos, acciones, etc., así como sus propiedades, el texto descriptivo de las acciones, los actores implicados o los monólogos de cada actor en los diálogos. Cada uno de estos elementos es analizado en una segunda fase, en la que cada uno es etiquetado con un conjunto de emociones básicas, a partir de las cuales se determinará la intensidad emocional. Estas dos primeras fases del sistema son realizadas en el componente *Analizador de Guiones* del módulo lógico.

Con esta estructura de datos representativa del guión, con las distintas escenas que componen la historia, los elementos de cada una, y las propiedades de éstos, se genera la estructura lógica de la historia antes mencionada, que guiará en tiempo real al *Director* para determinar la configuración de cámara más apropiada a la secuencia que se está desarrollando. Para ello, la aplicación genera una secuencia de acción para cada elemento de acción encontrado en el guión, así como una secuencia de diálogo para cada conjunto de monólogos de actores analizados en el guión, agrupando éstas según el orden temporal del guión en las escenas identificadas.

A su vez, el *Director* se vale de un conjunto de eventos, proporcionados por el módulo gráfico, que indicarán el comienzo o finalización de los movimientos o diálogos de los personajes, y que permitirán sincronizar la historia generada en tiempo real, así como especializar las secuencias en acciones estáticas o con movimiento, o diálogos estáticos o con movimiento. Además, se ha incluido otro tipo de evento que permite identificar cuándo un actor entra o sale de la escena actual, pudiendo de esta manera sincronizar escenas en las que los actores salgan de una localización para entrar en otra en la que continúa el flujo de la historia con ellos mismos.

Una vez que el *Director* ha determinado la mejor configuración de cámara a partir de la información proporcionada por la estructura de la historia y los eventos del módulo gráfico, se realiza el proceso de interpretación y ejecución de dicha configuración en el entorno virtual por parte del *Gestor de Cámaras* del módulo gráfico, el cual, como se ha dicho, ha sido adaptado para poder representar visualmente dichas configuraciones.

Una visión global de este proceso se puede ver en el Algoritmo 1, donde se muestran las distintas etapas que se realizan en la aplicación, y que han sido brevemente expuestas, dando una perspectiva general del proceso que realiza el sistema para controlar la gestión de las cámaras en tiempo real. A lo largo de las siguientes secciones se describirá en detalle cada fase, explicando los puntos más característicos de cada una, así como su función en el proceso global.

---

**Algoritmo 1** Algoritmo de Gestión de Cámaras Basado en Emociones

---

```
1:  $EDG \leftarrow$  Análisis del guión en HTML y creación de la Estructura de
   Datos del Guión ( $EDG$ )
2:  $EDG \leftarrow$  Cálculo y asignación de la intensidad emocional a las acciones,
   diálogos y texto de los actores en la estructura  $EDG$ 
3:  $EDH \leftarrow$  Generación de la Estructura de Datos de una Historia ( $EDH$ )
   en escenas y secuencias a partir de  $EDG$ 
4:  $ModuloGrafico \leftarrow$  Inicialización del módulo gráfico
5:  $FinHistoria \leftarrow$  false
6:  $ConfCamInicial \leftarrow$  Cálculo de la configuración de cámara inicial a
   partir de la primera secuencia en  $EDH$ 
7:  $ModuloGrafico \leftarrow$  Envío de la configuración de cámara
    $ConfCamInicial$  para su interpretación y representación en el
   entorno virtual
8: while  $FinHistoria \neq true$  do
9:    $Eventos \leftarrow$  Recepción de eventos
10:  if Hay algún evento then
11:     $CambioSecuencia \leftarrow$  Comprobar si hay un cambio de secuencia en
     $EDH$ 
12:     $ActualizarSecuencia \leftarrow$  Comprobar si hay que actualizar la se-
    cuencia en  $EDH$ 
13:  end if
14:  if  $ActualizarSecuencia$  o  $CambioSecuencia$  o la configuración de
    cámara ha agotado su tiempo then
15:     $FinHistoria \leftarrow$  Comprobar fin de la historia en  $EDH$ 
16:    if  $FinHistoria \neq true$  then
17:       $NuevaConfCam \leftarrow$  Cálculo de la nueva configuración con los
       $Eventos$  y la secuencia actual en  $EDH$ 
18:       $ModuloGrafico \leftarrow$  Envío de la nueva configuración de cámara
       $NuevaConfCam$  para su interpretación y representación en el
      entorno virtual
19:    end if
20:  end if
21: end while
```

---

#### 4.3.1. Fase I: Análisis del guión

Este sistema se basa principalmente en la estructura específica de los guiones, y en su análisis, para determinar los diferentes tipos de secuencias que han sido definidas en la aplicación, y extraer la información necesaria para seleccionar adecuadamente la configuración de cámara más apropiada. Para ello, como se ha dicho, el sistema debe reconocer los diferentes elementos que componen un guión, así como la localización en el texto de éstos, a partir de los cuales se creará una estructura de datos interna, que se irá enriqueciendo a lo largo del proceso de análisis con los diferentes parámetros analizados.

```
<p id="sb7G7000" class="sceneheading">INT. EDORAS,  
    GOLDEN HALL - NIGHT<br>  
</p>  
<p class="action">Eowyn sleeps in the centre  
    of the room next to the fire. Aragorn walks  
    past her to the fire. He moves the logs around  
    to encourage it to burn, he looks at her  
    sleeping and walks over to her to pull her  
    dress down to her ankle and her collar up  
    to her neck<br>  
</p>  
<p class="character">EOWYN<br>  
</p>  
<p class="parenthetical">(with her eyes closed)<br>  
</p>  
<p class="dialog">What time is it?<br>  
</p>  
<p class="character">ARAGORN<br>  
</p>  
<p class="dialog">Not yet dawn. <br>  
</p>  
<p class="action">Eowyn strokes the fur on her  
    collar and withdraws his hand. Eowyn grabs  
    his hand as he draws away. Aragorn turns back  
    and looks at her.<br>  
</p>
```

Figura 4.8: Formato HMLT exportado por la aplicación Celtx.

Para ello, la aplicación toma como entrada dos ficheros. Por un lado, el fichero con el texto del guión según el estándar (ver sección 2.2 del capítulo 2); y por otro, un fichero con la lista de los diferentes actores. El primero de estos ficheros está estructurado en el lenguaje de marcado HTML, a través del cual se puede identificar fácilmente los distintos elementos que componen el guión, así como su localización en el texto, utilizando un sistema básico de análisis con cualquier librería en Java para procesamiento de lenguajes de marcado, y permitiendo extraer la información necesaria de una forma sencilla. Un ejemplo de estos ficheros se puede ver en la Figura 4.8, don-

de se aprecian fácilmente los distintos elementos que componen un guión. Este formato de fichero es producido por el programa de edición de guiones de código abierto **Celtx**<sup>5</sup>, diseñado específicamente para la realización y gestión de guiones, facilitando enormemente la creación de éstos según el formato estándar. Este sistema ofrece también la posibilidad de exportarlos al formato HTML utilizado en este sistema, así como su importación desde texto plano.

El segundo de los ficheros, la lista con los diferentes actores que componen la historia, es utilizada para identificar en cada elemento cuáles son los personajes que intervienen en la secuencia. Se han evaluado distintos métodos para el reconocimiento automático de los nombres de los personajes, pero debido a su gran variedad, y a que en muchas ocasiones son inventados, los resultados obtenidos no fueron satisfactorios.

Con estas dos entradas, el componente *Parser de Guiones*, del módulo lógico del sistema, crea una estructura de datos representativa del guión (EDG), organizando en orden cronológico, las escenas, junto con el texto de la localización de éstas, así como sus distintos elementos, acciones y diálogos, identificando en ambos casos los actores que lo componen, el texto descriptivo en los primeros y los monólogos de los actores en los segundos. Los paréntesis, otro de los elementos de los guiones, y que representan pequeñas anotaciones utilizadas para describir la forma, principalmente emotiva, en que los actores interpretan los monólogos, son utilizadas en este sistema para enriquecer el proceso de reconocimiento de emociones de los textos de los diálogos de cada actor, por lo que son añadidos a los monólogos de éstos en el proceso de análisis del guión.

Un ejemplo de esta estructura de datos en formato XML se puede ver en la Figura 4.9, donde se muestran los distintos elementos del guión analizado, así como la jerarquía y representación de los mismos y sus atributos. De esta manera, esta primera etapa permite identificar la información necesaria dentro del guión de la historia, para, junto con los posteriores análisis, poder identificar correctamente cada tipo de secuencia en la historia y poder generar así la estructura de la historia que permita gestionar el sistema de cámaras del entorno virtual.

---

<sup>5</sup><http://www.celtx.com/>



```

<ScreenPlay>
  <Scene>
    <Text>INT. EDORAS, SLEEPING QUARTERS - NIGHT</Text>
    <Action actors="GIMLI,ARAGORN,PIPPIN">
      <Text>Gimli is snoring in the room. The floor of
        the hall is scattered with sleeping bodies.
        Aragorn moves stealthily across the room and
        goes out through the door. Pippin turns over and
        wakes up. He tries to sleep obviously having an
        inward struggle with himself.</Text>
    </Action>
  </Scene>
  <Scene>
    <Text>INT. EDORAS, GOLDEN HALL - NIGHT</Text>
    <Action actors="ARAGORN,EOWYN">
      <Text>Eowyn sleeps in the centre of the room
        next to the fire. Aragorn walks past her to the
        fire. He moves the logs around to encourage it
        to burn, he looks at her sleeping and walks
        over to her to pull her dress down to her
        ankle and her collar up to her neck</Text>
    </Action>
    <Dialog actors="EOWYN,ARAGORN">
      <Speech>
        <Actor>EOWYN</Actor>
        <Text>(with her eyes closed)What time is
          it?</Text>
      </Speech>
      <Speech>
        <Actor>ARAGORN</Actor>
        <Text>Not yet dawn.</Text>
      </Speech>
    </Dialog>
  </Scene>
</ScreenPlay>

```

Figura 4.9: Estructura de datos del guión en formato XML.

#### 4.3.2. Fase II: Cálculo de la intensidad emocional

Una vez generada la estructura de datos del guión (EDG) en la primera fase del algoritmo, se continúa el proceso de análisis, en esta segunda etapa, con el etiquetado de emociones de los textos que componen cada elemento, a través del cual se podrá determinar la intensidad emocional. Este segundo paso del algoritmo se vale de la información extraída en la fase anterior, añadiendo la información adicional de la intensidad emocional a cada elemento de la estructura.

Como ya se ha dicho, la intensidad emocional inmersa en la secuencia que se está filmando es una de las características principales de las técnicas cinematográficas, pudiendo determinar a partir de ella con mucha mayor precisión la mejor configuración de cámara. Para ello, en este trabajo se plantea el uso de la aplicación **Emotag** (Francisco y Hervás, 2007), imple-

mentada en la Universidad Complutense de Madrid, y descrita en mayor detalle en la sección 3.3.3, que permite etiquetar automáticamente un texto con un conjunto de emociones básicas, a través de las cuales se pueden determinar, en mayor o menor medida dependiendo del caso, las emociones inmersas en un texto.

Más concretamente, **Emotag** permite calcular la probabilidad de cada emoción, de entre un conjunto de emociones básicas, con la que sería etiquetado un texto, utilizando para ello aprendizaje automático sobre un *corpus* de cuentos etiquetado por humanos y diferentes técnicas de Procesamiento de Lenguaje Natural. El proceso de análisis de **Emotag** comienza por el reconocimiento de cada palabra del texto, determinando su categoría gramatical y su lexema, y eliminando del análisis todas aquellas palabras que aparecen en la lista de parada del programa; es decir, aquellas que no son *nombres*, *verbos*, *adverbios* o *adjetivos*. Con estos pares, la aplicación consulta la lista LEW aprendida (*List of Emotional Words*), la cual almacena las probabilidades de cada emoción para cada par, recuperando las probabilidades de las emociones almacenadas para ese par si éste existe en la lista. Si no aparece el par en la lista LEW, se procede a la consulta de los hiperónimos de la palabra en *WordNet*, comprobando para cada uno su ocurrencia de nuevo en la lista LEW, y recuperando sus datos en caso de existencia. Las probabilidades de cada emoción no básica en cada palabra son sumadas, según la jerarquía impuesta por la ontología del programa OntoEmotions, obteniendo así la probabilidad final de cada emoción básica para todo el texto.

Tras este análisis, **Emotag** proporciona como salida la emoción con mayor probabilidad de ser asignada al texto. En el caso de este sistema, se ha transformado esa salida para poder obtener las probabilidades de todas las emociones básicas, evitando así la pérdida de información en el cálculo de la intensidad emocional.

Todo este proceso, así como la interacción con el programa **Emotag**, es realizado por el componente *Analizador de Emociones* del módulo lógico del sistema. Para ello, recibe como entrada la estructura de datos generada en el paso anterior, analizando para cada acción el texto descriptivo, y para cada diálogo, los monólogos de los actores, y asignando a cada uno la intensidad calculada.

Cada conjunto de probabilidades de las emociones básicas de cada elemento devuelto por **Emotag** es normalizado en el intervalo  $\{0, 1\}$ , restringiendo así éstos a porcentajes entre 0 y 100. Este conjunto de emociones básicas, basado en diferentes teorías psicológicas expuestas en la sección 3.3.1, se puede ver en la Figura 4.10, donde a cada una de las emociones le ha sido asignado un peso en el rango  $\{-1, 1\}$ , dependiendo de la carga emocional inmersa en cada una. Como se expuso en la sección 3.3.1, existen muchas teorías psicológicas sobre las emociones humanas, pero desde el punto de vista de la intensidad, lo que sí parece ampliamente aceptado es que las emocio-

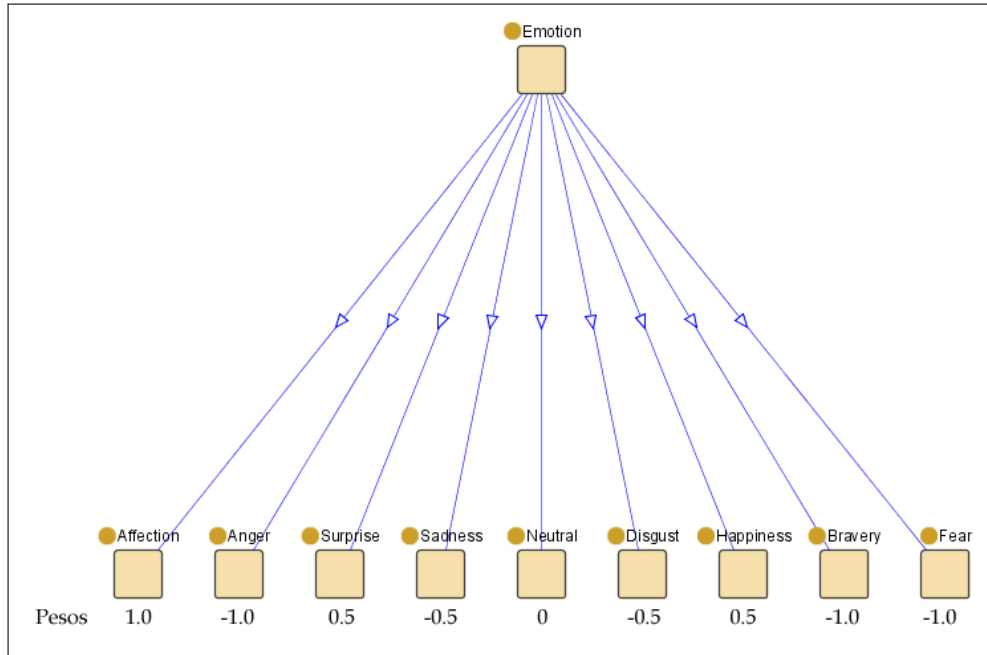


Figura 4.10: Emociones básicas manejadas por **Emotag**.

nes poseen una carga positiva o negativa, mayor o menor dependiendo de la emoción. Un ejemplo de esto se puede ver en el modelo de Watson y Tellegen (1985) expuesto en esa misma sección, donde se realiza una división circular de dos dimensiones sobre la afectividad positiva y negativa de las emociones, y donde, además, cada una posee una intensidad alta o baja. Apoyándose en esta teoría, y en su clasificación de las distintas emociones según la intensidad afectiva positiva y negativa, se ha realizado esta asignación inicial de pesos de las distintas categorías emocionales devueltas por **Emotag**.

Atendiendo a esto, una carga emocional positiva, representativa de sentimientos afectivos o de felicidad, es definida en el sistema con valores en el rango  $\{0,1\}$ , donde 0 representa una carga neutral y 1 es la máxima intensidad positiva. Por el contrario, una intensidad emocional negativa, representativa de emociones como el miedo o la tristeza, es definida con valores en el rango  $\{-1,0\}$ , donde 0 representa una carga neutral, como ya se ha dicho, y -1 indica la máxima carga negativa definida en el sistema.

A partir de este conjunto de pesos asignados a cada emoción, se puede deducir la intensidad emocional inmersa en un texto mediante la ecuación 4.1 propuesta, donde la intensidad emocional de cada elemento es calculada como la suma de todas las probabilidades del conjunto básico de emociones, ponderada cada una por el peso asignado. En este proceso, la emoción neutral ha sido designada con un peso cero, ya que se ha considerado pre-

ferible que todo texto tenga asociada una intensidad diferente a neutral, y también debido a que ésta puede darse en el sistema como la ausencia de probabilidades en el resto de emociones. El valor devuelto por la ecuación es normalizado de nuevo entre los valores  $\{-1, 1\}$ , obteniendo de esta manera un número en cada elemento que designará la carga emocional inmersa en el texto, pudiendo ser ésta, positiva, neutral o negativa, como se ha definido anteriormente.

$$IntEmo = \sum_{i=0}^n EmoB_i * P_i \quad (4.1)$$

En resumen, el proceso del calculo de la intensidad emocional para cada elemento se puede dividir en los siguientes pasos:

1. Análisis del texto, ya sea la descripción de las acciones o el monólogo del actor en los diálogos, por el programa **Emotag**, identificando las emociones básicas y sus probabilidades de ser asignadas a ese texto.
2. Normalización de las probabilidades de todas las emociones básicas devueltas por **Emotag** para ese elemento.
3. Aplicación de la formula 4.1 para el calculo de la intensidad emocional de cada elemento mediante los pesos establecidos en el sistema para cada emoción básica.
4. Normalización del resultado de la intensidad calculada entre los valores  $\{-1, 1\}$ .
5. Asignación a la estructura de datos del guión de la intensidad de cada elemento. En el caso de los diálogos, cada monólogo es etiquetado con una intensidad emocional, a partir de las cuales se genera la intensidad del diálogo global, mediante la suma de las intensidades de éstos y su posterior normalización.

De esta manera, la estructura creada en el primer paso del algoritmo es enriquecida con la información emocional generada por el sistema, haciendo uso para ello del análisis realizado por el programa **Emotag**, y obteniendo así la información suficiente para poder determinar en tiempo real la configuración de cámara más adecuada. Como se puede ver en la Figura 4.11, a cada acción de la estructura previa se le ha asignado la intensidad emocional calculada en esta segunda fase, así como a los monólogos de cada actor, y a los diálogos que engloban a éstos.

```

<ScreenPlay>
  <Scene>
    <Text>INT. EDORAS, SLEEPING QUARTERS - NIGHT</Text>
    <Action emotionalIntensity="-0.16299790356394123"
    actors="GIMLI, ARAGORN, PIPPIN">
      <Text>Gimli is snoring in the room. The floor of
        the hall is scattered with sleeping bodies.
        Aragorn moves stealthily across the room and
        goes out through the door. Pippin turns over
        and wakes up. He tries to sleep obviously
        having an inward struggle with himself.</Text>
    </Action>
  </Scene>
  <Scene>
    <Text>INT. EDORAS, GOLDEN HALL - NIGHT</Text>
    <Action emotionalIntensity="0.32771009059668854"
    actors="ARAGORN, EOWYN">
      <Text>Eowyn sleeps in the centre of the room
        next to the fire. Aragorn walks past her to
        the fire. He moves the logs around to encourage
        it to burn, he looks at her sleeping and walks
        over to her to pull her dress down to her ankle
        and her collar up to her
        neck</Text>
    </Action>
    <Dialog emotionalIntensity="0.3286863480162449"
    actors="EOWYN, ARAGORN">
      <Speech emotionalIntensity="0.3240393626991565">
        <Actor>EOWYN</Actor>
        <Text>(with her eyes closed)What time is
          it?</Text>
      </Speech>
      <Speech emotionalIntensity="0.3333333333333333">
        <Actor>ARAGORN</Actor>
        <Text>Not yet dawn.</Text>
      </Speech>
    </Dialog>

```

Figura 4.11: Estructura de datos del guión enriquecida con la intensidad emocional.

#### 4.3.3. Fase III: Creación de la historia virtual

Una vez completados los procesos de extracción y análisis de la información relevante de la historia que se va a visualizar, se realiza el proceso, en esta tercera etapa del algoritmo, de creación de la estructura de datos de la historia (EDH) que guiará al *Director*, junto con los eventos producidos en el mundo virtual, a través de la historia para poder determinar la mejor configuración de cámara acorde a lo que está sucediendo en tiempo real.

Para ello, como se ha dicho, las diferentes acciones de la historia son

clasificadas en los dos tipos de secuencias existentes en la aplicación, según sean de diálogo o de acción. Esto es fácilmente identificable en la estructura generada en los pasos previos del algoritmo, donde cada elemento del guión ha sido identificado y estructurado en un orden cronológico, junto con los datos más importantes para el correcto tratamiento de la gestión de cámara.

Para poder realizar esta tarea, el *Director*, se vale de un simple mapeo entre los distintos elementos de la estructura del guión (EDG) y las secuencias que componen una historia, generando una nueva estructura formada por las escenas existentes en el guión, ordenadas cronológicamente, cada una compuesta por las distintas secuencias, ordenadas también cronológicamente, resultantes del mapeo de los distintos elementos. De esta manera, cada elemento de acción de la estructura del guión se traduce a una secuencia de acción dentro de la estructura de la historia. De la misma manera, cada elemento de diálogo en la estructura del guión es traducido como una secuencia de diálogo, donde cada monólogo de cada actor es almacenado como un par formado por nombre del actor y la intensidad emocional de su diálogo, para identificar dentro de la secuencia quién está hablando y su carga emocional.

```
<Story>
  <Scene>
    <ActionSeq emotionalIntensity="-16.299790356394123"
      motion="false" actors="GIMLI, ARAGORN, PIPPIN"
      actorsInMotion="">
    </ActionSeq>
  </Scene>
  <Scene>
    <ActionSeq emotionalIntensity="32.771009059668856"
      motion="false" actors="ARAGORN, EOWYN"
      actorsInMotion="">
    </ActionSeq>
    <DialogSeq emotionalIntensity="32.86863480162449"
      motion="false" actors="EOWYN, ARAGORN"
      actorsInMotion="">
      <Speech>
        <Actor>EOWYN</Actor>
        <EmotionalIntensity>
          32.403936269915654</EmotionalIntensity>
        </Speech>
        <Speech>
          <Actor>ARAGORN</Actor>
          <EmotionalIntensity>
            33.33333333333333</EmotionalIntensity>
          </Speech>
        </DialogSeq>
    </DialogSeq>
  </Scene>
</Story>
```

Figura 4.12: Estructura de datos de la historia en formato XML.

Como se puede ver en la Figura 4.12 de la estructura de la historia en formato XML, cada elemento del guión ha sido traducido a una secuencia,

junto con los datos analizados relevantes para la elección de las posiciones y movimientos de cámara dentro de ésta. Además, las secuencias precisan de otro tipo de información importante para su correcta gestión, como es la presencia o no de movimiento por parte de los actores. Para ello, cada secuencia posee el atributo *motion* que permite identificar en tiempo real esta característica, el cual es actualizado con los eventos proporcionados por el módulo gráfico, especializando así las secuencias, tanto de diálogo como de acción, en estáticas o con movimiento, puesto que las diferentes técnicas cinematográficas difieren según este parámetro. Para una mejor gestión de estas técnicas específicas de movimiento, la secuencia posee otro parámetro, *actorsInMotion*, que permite identificar qué actores son los causantes del movimiento, y que es actualizado mediante los eventos de inicio de movimiento y finalización del movimiento de cada actor.

#### 4.3.4. Fase IV: Elección de la configuración de cámara

Esta etapa es, quizás, la más importante de todas, puesto que en ella se realiza el proceso de selección de la mejor configuración de cámara en tiempo real a partir de toda la información recopilada en las fases anteriores, y de eventos recogidos en el entorno virtual, los cuales permiten especializar las diferentes secuencias, y sincronizar el flujo de la historia según la estructura.

Para ello, esta cuarta etapa del algoritmo precisa de varios pasos para poder determinar la situación actual de la secuencia, a través de la estructura generada y los eventos recibidos del entorno virtual, así como discernir a partir de ésta en tiempo real, si es necesario un cambio de configuración de cámara o no, y si así fuera, cuál sería el más adecuado según lo que está sucediendo. Todo este proceso está encapsulado en el bucle de refresco del entorno virtual, en el que en cada iteración se comprueba si se han producido eventos nuevos, actualizando en caso afirmativo la secuencia que está sucediendo, y devolviendo una nueva configuración de cámara si fuera preciso.

De esta manera, el *Director*, en cada vuelta del bucle realiza el proceso de actualización de la secuencia que se está desarrollando, comprobando los eventos y determinando si se ha avanzado a la siguiente secuencia de la historia, o si se debe actualizar la secuencia actual especializándola en estática o con movimiento, o si se ha agotado el tiempo de la configuración de cámara actual. Si alguna de estas condiciones se cumpliera, el *Director* enviaría la secuencia actualizada al *Gestor de Cámaras*, el cual, a partir de ésta y de la base de reglas implementada, determinaría la nueva configuración de cámara más apropiada.

La primera de estas operaciones realizadas en cada iteración por el *Director* es el análisis de los eventos recibidos por parte del módulo gráfico. En este sistema se han introducido un conjunto de eventos, como ya se ha dicho, que permiten reconocer en tiempo real qué es lo que está sucediendo

en cada instante, y ayudar al *Director* a sincronizar la secuencia correcta de la estructura de la historia, así como la escena, a la vez que proporcionan información sobre su especialización. Para ello, se han diseñado eventos (consultar sección 4.2.2), que permiten identificar cuándo un actor comienza o finaliza una acción de habla o movimiento, lo que es fácilmente identificable gracias a la estructura de acciones del módulo gráfico mediante un simple mapeo de éstas. Además, se ha añadido otro evento que permite identificar cuándo un actor en movimiento produce un cambio de localización; es decir sale o entra en la escena, mediante la interacción de éste y los planos de localización del escenario del entorno virtual.

```
<events>
  <event actor="ARAGORN" type="INIT_MOTION"/>
  <event actor="ARAGORN" type="CHANGE_LOCATION"/>
  <event actor="ARAGORN" type="FIN_MOTION"/>
  <event actor="ARAGORN" type="INIT_MOTION"/>
  <event actor="ARAGORN" type="FIN_MOTION"/>
  <event actor="EOWYN" type="INIT_DIALOG"/>
  <event actor="EOWYN" type="FIN_DIALOG"/>
  <event actor="ARAGORN" type="INIT_DIALOG"/>
  <event actor="ARAGORN" type="FIN_DIALOG"/>
</events>
```

Figura 4.13: Eventos del sistema.

Un ejemplo de este tipo de eventos se puede ver en la Figura 4.13, donde se puede apreciar el formato y estructura de los mismos, y donde cada evento va asociado con el actor que lo ha producido, pudiendo así identificarlo en la secuencia. Esta comunicación entre los dos módulos del sistema, al igual que en el sentido contrario como ya se expuso la sección 4.2, es realizada mediante el lenguaje de marcado XML, tal y como se puede apreciar en la imagen. La elección de esta técnica se debe, principalmente, a que el intercambio de caracteres es una forma sencilla y poco costosa de interacción entre los dos lenguajes de programación utilizados mediante el sistema de conexión JNI, y a la capacidad de estructuración y reutilización que proporciona este lenguaje.

Una vez identificados los distintos eventos producidos en el entorno virtual, así como los actores a los que van asociados, se procede a la comprobación de la actualización de la secuencia actual. En este proceso se pueden dar tres tipos de circunstancias dependiendo de los eventos recibidos y de la secuencia actual, a partir de las cuales se determina si es necesario, o no, un cambio de secuencia o una actualización de la actual.

- **Actualización de la secuencia:** Este caso se produce cuando estando en una secuencia, ya sea de acción o de diálogo, se recibe un evento de inicio de movimiento, o de finalización de movimiento si ya



hubiera iniciado algún personaje previamente el movimiento, especializando así la secuencia actual en estática o con movimiento, según proceda. Para indicar esto, los parámetros *motion* y *ObjectsInMotion* de la secuencia son actualizados debidamente, permitiendo así que en un paso posterior el *Director* pueda proporcionar la nueva secuencia al *Gestor de Cámaras* y generar una nueva configuración cámara.

- **Cambio de secuencia por acción:** En este caso, por el contrario, el resultado final no produce una actualización de la secuencia actual, sino un cambio a la siguiente secuencia de la historia, obteniendo así la sincronización de la estructura de la historia con lo que está ocurriendo en el entorno virtual. Un ejemplo de esto se puede ver cuando se recibe un evento de inicio de diálogo y la secuencia actual que se está visualizando, según la estructura generada de la historia, es de acción. Debido a la estructura específica de los guiones, explicada en la sección 2.2, y a que el análisis de éstos preserva el orden cronológico de las acciones, en este caso, la siguiente secuencia a visualizar sería una secuencia de diálogo, la cual representaría la acción de diálogo que acaba de comenzar, debiendo de esta manera avanzar en el flujo de la estructura de la historia. De la misma manera, si se está en una secuencia de diálogo, y se recibe un fin de diálogo, y ya se han visualizado todos los monólogos de los actores en ese diálogo, se debe producir un cambio de secuencia, pudiendo ser la siguiente otra secuencia de diálogo o de acción.
- **Cambio de secuencia por cambio de localización:** En este caso, al igual que en el anterior, el resultado final produce un cambio de secuencia, y de la misma forma, una sincronización entre la estructura de la historia y el entorno virtual. Sin embargo, las condiciones que producen esta situación se basan en el cambio de localización de los actores, es decir, en la entrada y salida de éstos de la escena actual, la cual es especificada mediante las acciones en los guiones. Puesto que la estructura generada preserva el orden cronológico de la historia, en estos casos, es preciso un cambio de la secuencia actual, la cual estará indicando la acción del actor que sale o entra de la escena, siempre y cuando la siguiente secuencia no sea de diálogo, cuyo cambio estaría indicado por un evento de inicio de diálogo. De esta manera, se cubren situaciones en las que, por ejemplo, un actor sale de la escena y el flujo de la historia continúa con él en otra escena diferente. Al realizar este cambio, la estructura de la historia determinará que se ha acabado la escena, al estar visualizando su última secuencia, y continuará la historia en la escena siguiente.

Tras este proceso de análisis, se lleva a cabo, si fuera necesario, la actualización de la estructura de la historia, avanzando a la siguiente secuencia,

comprobando si ha finalizado la escena actual y es preciso cambiar a la siguiente, o si ha finalizado la historia completa.

Si la historia no ha finalizado y se ha producido un cambio en la secuencia, ya sea porque se ha actualizado la actual o porque se ha avanzado a la siguiente, o el tiempo de la configuración de cámara actual ha expirado, se debe seleccionar una nueva configuración de cámara acorde a los nuevos acontecimientos. Para ello, el *Director* proporciona la secuencia actual, modificada o no, al *Gestor de Cámaras*, el cual determinará la nueva configuración de cámara más adecuada a lo que está sucediendo mediante el conjunto de reglas implementado.

Para poder realizar este último paso de esta etapa, el *Gestor de Cámaras* se vale de la secuencia, que conlleva toda la información relevante de lo que está sucediendo para la elección de la mejor configuración de cámara, y del sistema de reglas, expuesto en mayor detalle en la sección 4.1.2 y en el Apéndice A. Así, a través de los diferentes parámetros de la secuencia, como los actores implicados, la intensidad emocional, la especialización de la secuencia en estática o con movimiento, o los actores implicados en el movimiento, y la configuración de cámara previa, puesto que en muchos casos la correcta realización de las técnicas cinematográficas precisa de varias configuraciones de cámara, se puede determinar la mejor manera de mostrar lo que está sucediendo en la historia.

```
<CameraConfiguration>
  <Movement type="PAN">
    <position>LONG_SHOT</position>
  </Movement>
  <Angle value="10"/>
  <PrincipalObjects>
    <Object>GIMLI</Object>
    <Object>ARAGORN</Object>
    <Object>PIPPIN</Object>
  </PrincipalObjects>
</CameraConfiguration>
<CameraConfiguration>
  <Distance value="MEDIUM_FULL_SHOT"/>
  <Angle value="0.0"/>
  <PrincipalObjects>
    <Object>GIMLI</Object>
  </PrincipalObjects>
</CameraConfiguration>
```

Figura 4.14: Configuraciones de cámara seleccionadas por el sistema.

Un ejemplo de este tipo de configuraciones devueltas por el *Gestor de Cámaras* se puede observar en la Figura 4.14, donde se muestra la estructura en formato XML de algunas de las configuraciones de cámara para el guión del ejemplo del primer paso. Estas configuraciones, como se puede ver,

son independientes del sistema virtual utilizado, abstrayendo así a un nivel superior la gestión de las cámaras, y relegando en el adaptador del entorno virtual el aspecto geométrico característico de cada sistema, permitiendo de esta manera una mejor y más fácil gestión y diseño de nuevas reglas.

Así, las configuraciones generadas por el sistema están formadas por diferentes elementos representativos de las distintas técnicas cinematográficas expuestas en el capítulo 2, como son la distancia, para la que se han elegido los 9 tipos expuestos en la sección 2.3, el ángulo de inclinación de la cámara con respecto al actor, o los movimientos. Estos últimos, han sido diseñados en el sistema como objetos diferentes dependiendo del tipo, como se expuso en la sección 2.8, donde cada uno posee parámetros específicos para su gestión por parte del adaptador del entorno virtual, como es, por ejemplo, la distancia a la que debe estar la cámara en el movimiento *Pan*, o la distancia inicial y final en el movimiento *Zoom*.

Con este resultado final, en cada iteración del bucle, el módulo lógico selecciona, si fuera necesario, la configuración de cámara del conjunto de reglas más adecuada, relegando al módulo gráfico y a su *Gestor de Cámaras*, las distintas operaciones geométricas necesarias para poder realizar las posiciones y movimientos de las cámaras seleccionadas.

#### 4.3.5. Fase V: Representación gráfica de la configuración de cámara

Dichas configuraciones son enviadas por el *Director* al *Gestor de Cámaras* del módulo gráfico en formato XML, el cual, mediante su sistema de *parser* es capaz de identificar los diferentes elementos de la configuración para su posterior ejecución. El adaptador se ha implementado de tal manera que pueda ejecutar las diferentes configuraciones de cámara existentes, de tal forma que pueda ejecutar los diferentes planos existentes, la orientación de la cámara respecto del ángulo vertical o los movimientos *Pan* y *Zoom*.

Además, las configuraciones de cámara aportan la información de los objetos sobre los que se deben realizar las distintas técnicas, por lo que el gestor debe diferenciar si el movimiento o plano a realizar va a ejecutarse sobre uno, dos o más personajes, puesto que la disposición de la cámara en cada configuración difiere en función del número de objetos. Otro de los problemas a los que se enfrenta este gestor es que los personajes en los entornos virtuales suelen tener el centro de coordenadas de sus modelos en posiciones diferentes a la cabeza, por lo que el resultado obtenido no sería el adecuado. Para solventar esto, el módulo gráfico asigna a cada objeto una *Bounding Box*, o caja invisible, situada exactamente en el lugar que se desea, en este caso las cabezas de los personajes, y que será el centro de referencia de las coordenadas del personaje, u objeto, para el *Gestor de Cámaras*.

Con todo esto, una vez reconocidos los parámetros de la configuración de cámara, el gestor discrimina entre si se debe realizar un movimiento so-

bre algún objeto, o por el contrario, una posición estática de la cámara. En el caso de las configuraciones estáticas se pueden diferenciar tres representaciones dependiendo del número de personajes principales, variando desde las configuraciones para un objeto, dos, o tres o más objetos. La primera de ellas es la más sencilla de todas, ya que se debe realizar una configuración estática, con el tipo de plano indicado, sobre un solo objeto, lo que sólo supone calcular la dirección de orientación del objeto, hacia donde mira, y colocar la cámara en esa dirección a la distancia requerida pero orientada en sentido contrario. Este proceso es refrescado en cada iteración para ajustarse correctamente aun incluso si el personaje se está desplazando. En el caso de la configuración de dos personajes el proceso es muy similar, sólo que en este caso el punto a donde debe apuntar la cámara es la posición intermedia entre ambos actores. La posición de la cámara se calcula trazando una recta que pase por el punto medio y perpendicular a la recta que une a los actores, situándola a la distancia indicada y orientada hacia esa posición.

Las configuraciones para tres o más personajes son un poco más complejas ya que se debe calcular el centro del polígono que forman los personajes u objetos. Para poder determinar esto se optado por la Formula 4.4, utilizada en sistemas de información geográfica, que permite calcular el centroide de un polígono. Para ello los vértices del polígono, en este caso las posiciones de los personajes, deben estar ordenados en el sentido de las agujas del reloj, por lo que se ha implementado el algoritmo *Andrew's Monotone Chain Convex Hull*, el cual a partir de un conjunto de puntos en el espacio devuelve el polígono convexo que forman los puntos, estando éstos puntos orientados según el sentido de las agujas del reloj. De esta manera, el sistema es capaz de determinar la posición central del polígono, orientando la cámara hacia ella y calculando una perpendicular a uno de los lados del polígono para posicionarla a la distancia requerida.

$$A = (1/2) \sum_{i=0}^{n-1} (x_i * y_{i+1} - x_{i+1} * y_i) \quad (4.2)$$

$$C_x = (1/A * 6) \sum_{i=0}^{n-1} ((x_i + x_{i+1})(x_i * y_{i+1} - x_{i+1} * y_i)) \quad (4.3)$$

$$C_y = (1/A * 6) \sum_{i=0}^{n-1} ((y_i + y_{i+1})(x_i * y_{i+1} - x_{i+1} * y_i)) \quad (4.4)$$

Para este trabajo se han implementado los movimientos de *Pan* y *Zoom*, que al igual que en el caso estático, se pueden resolver para uno, dos, o tres o más personajes. En el caso del *Pan* la configuración para una persona es similar a la estática. La cámara se posiciona a cierta distancia si ésta

está indicada en la configuración, o si no en la posición en la que se encuentra en ese momento, realizando un barrido sobre el movimiento del personaje indicado. Para el caso de dos o más personajes, el barrido se realiza entre la distancia horizontal existente entre los personajes proporcionados por la configuración, ya estén en movimiento o estáticos.

En el movimiento *Zoom* el proceso es similar al estático para todas las configuraciones de personajes, diferenciándose en que a partir de la ejecución de la configuración, en cada iteración la distancia se va reduciendo una constante asignada hasta la distancia indicada en la configuración de cámara proporcionada por el módulo lógico, produciendo un acercamiento progresivo hacia los objetos.

En todos los casos expuestos, el ángulo indicado en la configuración es calculado para la correcta asignación de la orientación de la cámara respecto del eje vertical.

De esta manera el *Gestor de Cámaras* del entorno virtual es capaz de representar las configuraciones de cámara seleccionadas por el módulo lógico acordes a lo que está sucediendo y que representan las diferentes técnicas cinematográficas presentadas. Un ejemplo de su ejecución se mostrará en el Capítulo 5.



---

## Capítulo 5

# Resultados y evaluación

Hasta este punto se han definido todos los componentes del sistema que intervienen en el algoritmo automático de dirección cinematográfica, así como las interacciones entre ellos y las distintas aplicaciones utilizadas en este trabajo. A lo largo de esta sección, se mostrará un ejemplo del funcionamiento del sistema, exponiendo los diferentes logros obtenidos e identificando los fallos encontrados.

### 5.1. Simulación de un fragmento de una película

A lo largo de toda esta sección se mostrarán los resultados obtenidos en la simulación de un fragmento de la película “El Señor de los Anillos: El retorno del Rey”, la cual posee gran cantidad y calidad de diferentes técnicas cinematográficas, siempre vistas desde la perspectiva e interpretación de un director humano, que proporcionan un interesante punto de referencia con respecto al cual evaluar los resultados conseguidos. Ambas secuencias se compararán exponiendo el por qué de las distintas similitudes y diferencias existentes entre ellas.

El fragmento de la película seleccionado para la simulación y evaluación del funcionamiento del sistema desarrolla la acción en el Castillo de Rohan, cuando el personaje de *Aragorn* se encuentra de pie y dispuesto para salir de una de las habitaciones del Castillo, en la que todos los miembros de la Comunidad del Anillo permanecen dormidos menos *Pippin*. Esta secuencia continúa en el salón principal del Castillo, hacia donde se dirige *Aragorn* tras salir de la habitación, y en la que éste mantiene una conversación con *Eowyn*, tras la cual finaliza la escena con la salida de *Aragorn* del Castillo. El fragmento completo del guión utilizado en esta simulación se puede consultar en el Apéndice B, y que ha sido obtenido de la página *The Internet Movie Script Database*<sup>1</sup>.

---

<sup>1</sup><http://www.imsdb.com/>

La elección de este fragmento se debe al hecho de que estas dos escenas comprenden una amplia gama de situaciones que permiten una correcta evaluación del sistema propuesto, como son el diálogo entre *Aragorn* y *Eowyn*, la salida de un personaje de la escena, en el caso del personaje de *Aragorn* al salir de la habitación, o el movimiento de éste entre ambas escenas, pudiendo de esta manera comparar los resultados obtenidos con la película original.



Figura 5.1: Aragorn y Eowyn

Debido a que los modelos de los personajes que se han podido obtener para esta simulación no poseen todas las animaciones y rasgos fisionómicos que se representan en la película, y a que la distribución del Castillo es ligeramente diferente, no se ha podido generar una simulación idéntica a la original, como por ejemplo cuando *Aragorn* enciende la pipa en la hoguera del salón del Castillo, aunque esto no supone un problema a la hora de realizar una correcta evaluación de los resultados obtenidos. Así mismo, se hace preciso, para el adecuado entendimiento del ejemplo, una identificación





Figura 5.2: Gimli y Pippin

previa de los distintos modelos utilizados en la simulación con sus correspondientes personajes de la película, tal y como se muestra en las Figuras 5.1 y 5.2.

Las diferentes acciones realizadas por los actores en la simulación han sido sincronizadas según los tiempos de la película, e intentando ser lo más similares a ésta en cuanto a posiciones y movimientos de los actores, siempre teniendo en cuenta que en un entorno virtual no existen cortes temporales y espaciales. Además, se ha extraído el sonido de la película para hacer la simulación más real. La visualización completa de la simulación se puede ver en un video en la página web <http://nil.fdi.ucm.es/index.php/REC>.

Para poder realizar una valoración adecuada de los resultados obtenidos en la simulación, se propone una clasificación de tres situaciones que permiten identificar correctamente los beneficios y las debilidades del sistema,

y que serán descritas a lo largo de las siguientes secciones. Por un lado, se mostrarán aquellas secuencias cuya disposición de cámaras es similar a las de la película, y que desde la perspectiva de las técnicas cinematográficas son consideradas correctas. Se continuará exponiendo aquellas configuraciones que difieren de la película original pero que, desde el punto de vista de las técnicas de cine, responden a un comportamiento óptimo, concluyendo con las secuencias que son distintas de las de la película y que no responden al comportamiento correcto esperado.

### 5.1.1. Secuencias similares y correctas

Quizá la secuencia más similar de toda la simulación con respecto a la película original, y correcta desde el punto de vista cinematográfico, sea la escena en la que *Aragorn* y *Eowyn* mantienen la conversación. Tanto en la película como en la simulación, la escena comienza con una configuración presentando a los dos actores desde un plano largo, continuando en la película con un plano corto sobre la acción de *Aragorn* arrojando a *Eowyn*, y en la simulación, con un plano medio sobre éste. Las siguientes secuencias son casi idénticas, presentando ambas a los actores mientras hablan, diferenciándose en esto en que en la simulación se presenta desde un plano medio mientras que la película se realiza desde un plano corto. Una diferencia notable se produce al finalizar *Aragorn* su primer monólogo, en el cual la simulación vuelve a mostrar a ambos personajes desde un plano largo, mientras que en la película se muestra cómo éste inicia el movimiento de alejarse mientras *Eowyn* agarra su mano atrayéndole. Además, durante el monólogo más largo de *Eowyn*, en la simulación se intercalan dos enfoques sobre *Aragorn* y dos sobre *Eowyn*, algo similar a lo que sucede en la película, en la que sólo se realiza un cambio sobre *Aragorn*, acompañando el movimiento de éste al agacharse. La secuencia termina en ambos casos con un plano más alejado presentando a ambos. En las Figuras 5.3 y 5.4 se pueden ver algunas de las secuencias, tanto de la película como de la simulación, de esta escena, y aunque no se puede apreciar correctamente la semejanza entre ambas, sí se pueden detectar algunas de las diferencias y similitudes aquí expuestas.

Aunque ambas secuencias son bastante parecidas, resulta obvia la dificultad de conseguir una simulación exactamente igual a la película, ya que los sistemas automáticos no pueden a día de hoy detectar todas las peculiaridades de una escena a través del guión, junto con el hecho de que cada director imprime su propio sello a cada una de las secuencias. Además, como se ha dicho, la simulación no es idéntica y no contiene todos los movimientos de los personajes de la película.

Analizando más a fondo las diferencias, el caso concreto de la utilización de planos cortos en la película frente a los planos medios de la simulación se debe a que la intensidad emocional de la escena calculada no es tan alta como la interpretada por el director. Partiendo de que éste es un aspecto



bastante subjetivo, lo que sí es cierto es que el *corpus* sobre el que se basa el programa **Emotag** está compuesto por un conjunto de cuentos, lo que puede hacer que en ciertos análisis gran parte del vocabulario analizado no aparezca, ignorándolo o asignándole una emoción neutral. Es por esto que en algunos de los fragmentos del guión en los que quizá un humano asignaría una mayor intensidad, especialmente en el monólogo más largo de *Eowyn*, el programa genera una intensidad menor.

Como se ha dicho, en la simulación cuando *Eowyn* realiza su monólogo más extenso, la cámara enfoca dos veces a *Aragorn*. Este tipo de técnica se suele utilizar para dar mayor dinamismo a las conversaciones cuando el monólogo de un personaje es muy extenso, variando entre los distintos actores que intervienen. En el caso de la película, por el contrario, sólo se realiza un cambio, el cual acompaña la acción de agacharse de *Aragorn*, lo que muestra una interpretación propia del director de la escena, y que se podría conseguir en este sistema simplemente aumentando el tiempo de la configuración de cámara cuando un personaje está hablando. Al no realizarse la acción de agacharse de *Aragorn* en la simulación, ésta no es mostrada puesto que no se lanza ningún evento de movimiento, además de que en el guión de la película utilizado esta acción no es recogida.

La misma situación se presenta en acciones como aquella en la que *Aragorn* arropa a *Eowyn*, o tras los dos primeros monólogos, cuando *Eowyn* agarra a *Aragorn*, las cuales no han podido ser simuladas, aunque en estos casos sí que aparecen en el guión.

### 5.1.2. Secuencias distintas y correctas

Un ejemplo claro de este caso se puede ver en la secuencia en que *Aragorn* entra en el salón del Castillo donde se encuentra dormida *Eowyn*. Aunque la distribución de los elementos no es exactamente la misma, se pueden apreciar las diferencias en las técnicas utilizadas en la película respecto a la simulación, considerando ambas aproximaciones válidas desde el punto de vista cinematográfico. En la película, toda la secuencia está filmada por un movimiento horizontal de la cámara que muestra la habitación donde *Eowyn* se encuentra durmiendo y en la que *Aragorn* aparece por detrás para dirigirse a la hoguera del centro de la sala, acabando con un plano medio sobre éste. Por el contrario, en la simulación la secuencia comienza con un plano largo sobre ambos actores abriéndose según avanza *Aragorn* hacia la mesa, y continuando con movimiento de *Pan* sobre él hasta su posición final. En las Figuras 5.5 y 5.6 se muestran algunos planos de ambas secuencias, donde se pueden apreciar las diferencias aquí expuestas.

Si bien es cierto que la técnica utilizada en la película es perfectamente correcta, ya que a través del movimiento de la cámara se muestra toda la escena, pudiendo el espectador observar tanto el movimiento de *Aragorn* como la posición de *Eowyn* en la habitación, hasta finalizar con un plano medio de

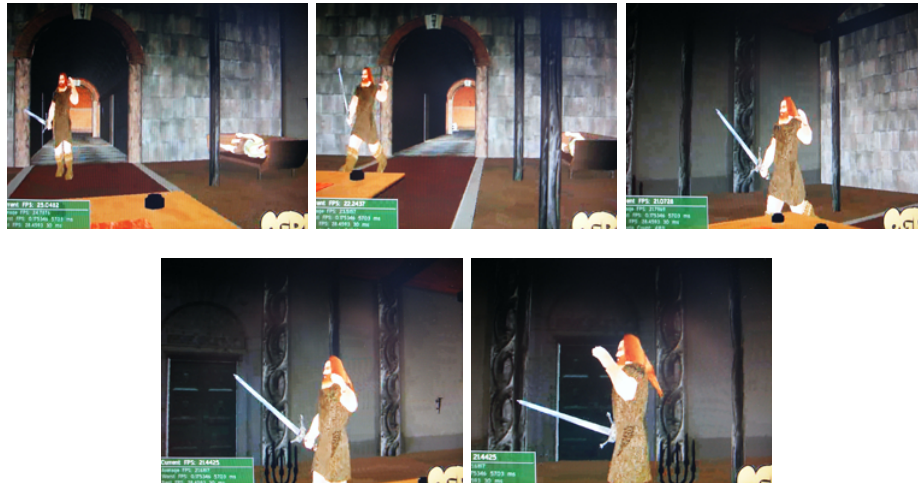


Figura 5.5: Secuencias de la simulación diferentes y correctas

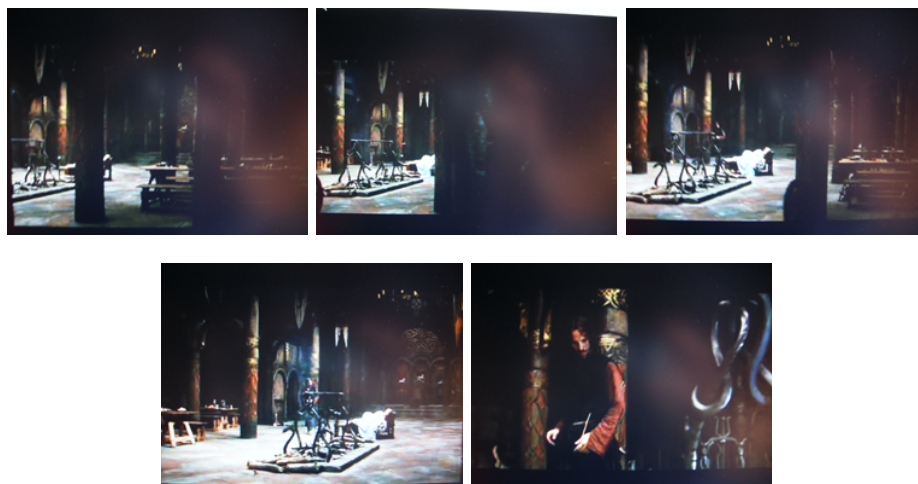


Figura 5.6: Secuencias de la película diferentes y correctas



*Aragorn* cuando éste ya está agachándose, esta secuencia denota una clara visión subjetiva del director de lo que está sucediendo, empezando por un plano casi oscuro para ir acercando al espectador a la acción completa de la habitación. Por el contrario, la simulación muestra, como en toda secuencia introductoria de una escena, un plano largo con los actores que componen dicha escena para orientar al espectador en la habitación y mostrarle lo que en ella ocurre, pero continuando en este caso con la acción principal de la escena, que es el movimiento de *Aragorn* hacia la mesa. Aunque ambas técnicas son igual de válidas, se podría decir que en la simulación se puede apreciar con mayor claridad el uso de técnicas más generales, mientras que la película muestra más una interpretación propia del director.

### 5.1.3. Secuencias distintas y erróneas

Este caso es propiciado por una de las principales debilidades del sistema y que se describirá con mayor detalle en la sección 5.2, intentando aportar posibles soluciones para trabajo futuro. Un ejemplo claro de esta situación se puede ver con claridad en las secuencias posteriores al movimiento de *Aragorn* hacia la mesa, en el caso de la simulación, o hacia la hoguera, en el caso de la película. En este último, la escena se centra mostrando a *Aragorn* manipulando un tronco de la hoguera, continuando con un plano largo sobre *Eowyn* tumbada, y terminando la secuencia con el movimiento de *Aragorn* hacia ésta con un plano largo. Por el contrario, en la simulación, cuando *Aragorn* llega a la mesa se continúa con un plano largo de ambos personajes en el que, debido a la distancia entre ellos, sólo aparece parcialmente *Aragorn*. Tras esto, se realiza un plano medio sobre cada uno de los personajes, continuando con otro plano largo de ambos en el que esta vez no se ve a ninguno de los personajes. La escena finaliza, al igual que en la película, con el movimiento de *Aragorn* hacia donde se encuentra *Eowyn* tumbada, pero con un movimiento de *Pan*. En la Figuras 5.7 y 5.8 se puede apreciar este caso claramente.

Estas secuencias son, quizá, las que mayores diferencias presentan con respecto a la película original, y ello se debe a diversos aspectos, como el hecho de que la distribución del castillo no sea igual, haciendo que el tiempo en el que el personaje de *Aragorn* anda sea menor en la simulación de lo que es en la película, generando una secuencia más larga tras el movimiento de éste en la que los actores están alejados. Además, el hecho de que *Aragorn* no realice la acción de agacharse a coger el tronco supone la utilización de técnicas diferentes, como es la visualización de ambos personajes en la escena, con un plano medio de cada uno tras esa localización inicial de la escena para el espectador. Sin embargo, esto produce el error antes mencionado, y es que aunque el plano asignado a esa secuencia es largo, la distancia entre ambos actores es mayor, por lo que, en un primer caso sólo se muestra parte de *Aragorn*, y en un segundo caso no se muestra a ninguno de los actores.

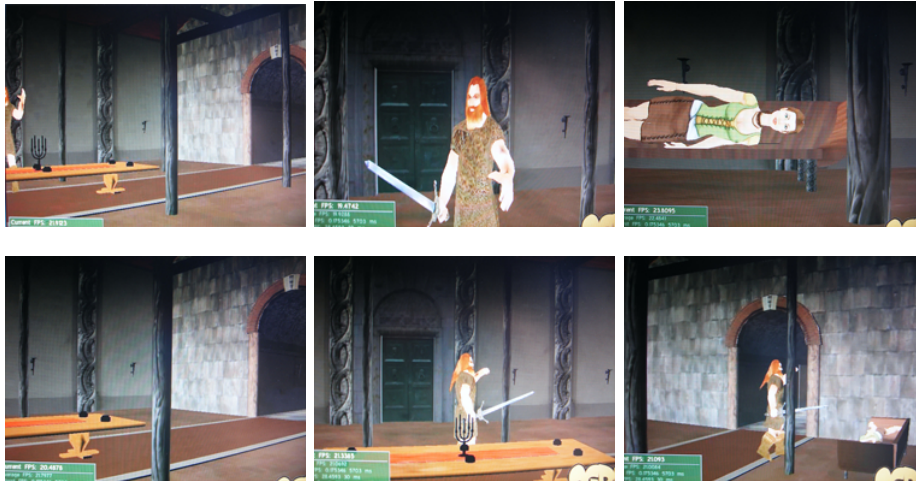


Figura 5.7: Secuencias de la simulación diferentes y erróneas



Figura 5.8: Secuencias de la película diferentes y erróneas

Esto se debe a que en la elección de la configuración de cámara a utilizar no se dispone de ningún tipo de información geométrica sobre los personajes y sus disposiciones, por lo que, al aplicar la regla, el resultado no es el esperado al no visualizar los elementos principales indicados en la configuración. Existen varias formas de solventar este tipo de problemas, las cuales serán descritas y propuestas como trabajo futuro en las siguientes secciones y en el Capítulo 6.

## 5.2. Discusión

Como se ha comentado no existen demasiadas investigaciones en este campo de los sistemas de gestión de cámaras basados en técnicas cinematográficas para entornos virtuales, y los pocos realizados no suelen presentar sistemas de evaluación o los resultados obtenidos, por lo que no se hace fácil una comparación entre los diferentes sistemas y el aquí planteado. Aun así, a lo largo de las dos siguientes secciones se van a mostrar las diferentes aportaciones e inconvenientes encontrados a lo largo del desarrollo del trabajo, intentando comparar, siempre que se pueda, con los trabajos previos analizados.

### 5.2.1. Principales aportaciones

Como ya se ha expuesto, la mayoría de los sistemas de gestión automática de cámaras basados en técnicas cinematográficas se centran principalmente en el aspecto geométrico de las distintas técnicas, intentando mediante ellas obtener una mejor calidad de las diversas situaciones existentes en la historia, y evitando problemas de oclusión o configuraciones de cámara donde la visualización de los personajes de la escena no sea del todo correcta, como por ejemplo conversaciones entre tres personajes en los que la cámara sólo muestra a dos y parcialmente al tercero (Christianson et al., 1996; wei He, Cohen, y Salesin, 1996; Bares, Grégoire, y Lester, 1998; Bares, Thainimit, y McDermott, 2000; Christie, Languénou, y Granvilliers, 2002; Christie y Languénou, 2003), sin tener en cuenta un aspecto tan importante como es la carga emotiva inmersa en las distintas técnicas cinematográficas, y que han dotado al cine de una gran riqueza visual.

Por el contrario, este trabajo se centra principalmente en el aspecto emotivo de las técnicas cinematográficas, aplicando a cada una de ellas los diferentes aspectos que la intensidad emocional de la escena proporciona, como por ejemplo el uso progresivo de planos más cortos con respecto a la intensidad emocional inmersa en el guión para situaciones de diálogo, o la utilización de movimientos de *Zoom*, variando las distancias también en relación a la intensidad, en vez de planos estáticos sobre personajes, cuando la intensidad emocional es alta en secuencias de acción.



Otro de los aspectos interesantes de este trabajo se centra en la utilización de técnicas de Procesamiento del Lenguaje Natural sobre el guión de la historia para obtener esta intensidad emocional. Si bien es cierto, como se ha dicho, que algunos trabajos previos añaden un factor emotivo a las técnicas cinematográficas utilizadas, (Tomlinson, Blumberg, y Nain, 2000; Amerson y Kime, 2001; Halper, Helbing, y Strothotte, 2001; Hornung, Lakemeyer, y Trogemann, 2003), la mayoría de ellos obtienen esta información mediante configuraciones realizadas a mano, o con plantillas preestablecidas sobre las acciones de los diferentes personajes, sin tener en cuenta la riqueza que proporciona la propia historia y el análisis de las emociones inmersas en ella. Gracias a la utilización del programa **Emotag**, el cual permite identificar las emociones inmersas en un texto, este sistema es capaz de obtener esa riqueza sin tener que realizar manualmente configuraciones previas sobre las acciones de los diferentes personajes, y pudiendo de esta manera pasar de un plano más abstracto, como es el lenguaje natural del texto del guión, a una disposición geométrica de posiciones y movimientos de las cámaras en la historia del entorno virtual.

Además, el análisis del formato del guión proporciona al sistema la capacidad de identificar los diferentes tipos de secuencias que componen la historia, que junto con los eventos del entorno virtual para la sincronización en tiempo real, permiten una mejor elección de la configuración de cámara más adecuada a la situación que está sucediendo en la historia en cada momento.

El sistema de reglas utilizado permite la propia interpretación de cada persona sobre las configuraciones de cámara utilizadas en cada situación, sin más que modificar las reglas, y sin la necesidad de realizar modificaciones de código, pudiendo de esta manera imprimir diversas visiones sobre una misma situación dependiendo del creador de la historia. Además, este sistema de reglas admite la utilización de lenguajes específicos de dominio (DSL), a partir de los cuales se pueden definir las reglas mediante una sintaxis similar al lenguaje natural (aunque esta mejora no ha sido implementada), permitiendo así una interfaz más sencilla para su modificación.

El hecho de excluir la información geométrica de la toma de decisiones sobre la configuración de cámara más adecuada, proporciona al sistema la capacidad de actuar de una forma genérica e independiente, pudiendo ser utilizado en cualquier entorno virtual con la simple tarea de adaptar su gestor de cámara para que interprete y ejecute las configuraciones propuestas por el módulo lógico, junto con la generación de un sistema de eventos respecto de las acciones descritas anteriormente.

### 5.2.2. Principales inconvenientes

Quizá una de las carencias, que desde otro punto de vista puede ser considerada una virtud, es la falta de conocimiento del sistema de dirección

cinematográfica sobre la disposición geométrica tanto de los actores como de los elementos existentes en la escena. Al abstraer a un nivel superior del geométrico las diferentes técnicas cinematográficas, se relega sobre la implementación del adaptador los diferentes cálculos geométricos de las posiciones finales de las cámaras. Aunque en un principio no supone un gran inconveniente, puesto que la mayoría de los sistemas gráficos actuales suelen disponer de una interfaz con numerosas funciones que facilitan este tipo de operaciones, como *tracking* sobre objetos, sistemas de coordenadas, sistemas de detección de colisiones, etc., es posible encontrar situaciones en las que las distintas operaciones realizadas para el correcto posicionamiento de la cámara según las directrices del *Director* no obtengan la forma óptima de visualizar la escena. Por ejemplo, se podría dar el caso de una escena en la que tres actores, muy separados unos de otros, se les deseará mostrar desde un plano medio con una posición apex (ver capítulo 2 sobre técnicas cinematográficas). En este caso, el sistema obtendría la visualización de uno, o a lo sumo dos de los actores, dependiendo de la distancia entre ellos. Lo mismo que ocurre en el ejemplo de la simulación mostrado en la sección 5.1.3.

Para solventar este tipo de situaciones se presentan dos posibles alternativas. Por un lado, se podría introducir el aspecto geométrico en la selección de la cámara más adecuada, lo que supondría un aumento del coste computacional, tanto en memoria como en tiempo de ejecución, obteniendo de esta manera una elección diferente para la cámara que no intentará mostrar a los tres actores en la escena a la vez. Un intermedio entre esta opción y la actual sería la utilización de otro tipo de eventos por parte del módulo gráfico, en la que se indicaran este tipo de situaciones, informando así al módulo lógico de que la configuración de cámara elegida no es óptima por colisión, o porque la disposición geométrica de los personajes no es la adecuada para el plano elegido.

En ambas aproximaciones sería interesante estudiar la posibilidad, en muchos aspectos más parecida a la función del director de cine, de que en estas situaciones el sistema, ya sea disponiendo de información parcial sobre la disposición de los actores o mediante configuraciones de éstos preestablecidas que el adaptador gráfico interpretará, pudiera desplazar los actores hasta una disposición más adecuada a las mejores técnicas, introduciendo planos que mostraran ese desplazamiento de los actores hasta la posición final deseada.

Otro de los inconvenientes encontrados en la realización de este sistema es la situación, todavía inicial, en la que se encuentran los distintos sistemas de reconocimiento de emociones existentes. Si bien es cierto que se ha realizado un gran avance, y que, desde la perspectiva de este autor, continuarán a grandes pasos en los próximos años, a día de hoy este tipo de sistemas no presentan un alto porcentaje de acierto a la hora de evaluarlos con cualquier tipo de texto. Aunque esto se presenta como una limitación del sistema, la utilización de este tipo de programas se considera que mejora, aún con las

deficiencias descritas, notablemente los resultados obtenidos.

Este sea, quizá, uno de los puntos débiles del programa **Emotag** de reconocimiento de emociones utilizado en el sistema, debido a que el *corpus* utilizado para el entrenamiento utiliza exclusivamente cuentos infantiles, lo que sin duda restringe mucho el vocabulario existente en la lista generada por el programa, y que será utilizada a la hora de evaluar los textos para etiquetarlos emocionalmente. Otro de los inconvenientes encontrados en este sistema es que en el proceso de análisis de los textos, el programa se basa sólo en las palabras y sus probabilidades, sumando el conjunto de todas para obtener la emoción final, pero sin tener en cuenta las relaciones entre ellas. Un claro ejemplo de este problema se puede ver en expresiones como “La muerte de la bruja” o la “La muerte del hijo”, en la que no se tendrían en cuenta las relaciones entre las palabras, asignando en ambos casos la misma probabilidad a la palabra muerte, sin tener en cuenta la relación con la palabra bruja, en el primer caso, o hijo, en el segundo.

Para intentar solventar esta deficiencia se podría estudiar la aplicación de la teoría de grafos a este tipo de sistemas, lo cual introduciría una capa más en el análisis en la que se tendrían en cuenta las distintas relaciones entre las palabras del *corpus*.

Otro aspecto que podría ser mejorado en el sistema es el cambio brusco producido entre algunas configuraciones seleccionadas por el módulo lógico, y que se puede apreciar en algunos casos en la simulación realizada en el sistema. Estos cambios son generados en el sistema al detectar un nuevo evento que produce una sincronización de la historia y sobre la que se debe volver a evaluar la configuración de cámara actual, proponiendo en ciertos casos transiciones bruscas entre éstas. Para intentar solventar esto se podrían introducir un conjunto de configuraciones intermedias entre ambas que produjeran una transición más suave desde las dos configuraciones seleccionadas como idóneas, generando así transiciones que no impactaran tanto al espectador, siempre y cuando no se deseara esto a la hora de visualizar la escena.



---

## Capítulo 6

# Conclusiones y trabajo futuro

Hasta este punto se ha descrito el sistema automático de dirección cinematográfica para entornos virtuales basado en emociones, el cual, a partir del guión de la historia que se va a visualizar, junto con los nombres de los actores y los eventos producidos en el entorno virtual adaptado al algoritmo, es capaz de seleccionar la mejor configuración de cámara de acuerdo a lo que está sucediendo en la escena en cada momento.

Además de esta descripción de los distintos componentes del sistema, y de sus diferentes funciones en el algoritmo, se han analizado los diversos trabajos previos encontrados en este campo y en los que esta investigación se ha apoyado, intentando aportar mejoras a las deficiencias encontradas en los mismos mediante las diversas técnicas expuestas a lo largo de esta memoria.

Para finalizar, se ha realizado una breve discusión sobre las diferentes aportaciones encontradas en el resultado final del proyecto con respecto al estado del arte actual de este tipo de sistemas, mostrando así mismo los diferentes inconvenientes encontrados.

Una vez realizado todo esto, sólo queda exponer el estado final del trabajo, así como extraer las conclusiones oportunas sobre él, describiendo además las futuras líneas de investigación que pretenden mejorar y solventar los diferentes problemas identificados.

### 6.1. Estado actual del trabajo

Una vez desarrollado el sistema, expuestos sus resultado, y habiéndolos evaluado, se puede concretar que el trabajo aquí presentado ha conseguido los objetivos marcados, como eran la introducción del aspecto emotivo en las diferentes técnicas cinematográficas para la gestión de la cámara, y la determinación de si el uso de técnicas de Procesamiento del Lenguaje

Natural sobre el guión de la historia podían proporcionar la información suficiente para este cometido, permitiendo además identificar y clasificar las diferentes secuencias que componen cada escena para una mejor elección de la configuración de cámara más acorde a lo que está sucediendo.

Aunque es cierto que en este primer prototipo se han detectado problemas que deben ser resueltos en un trabajo futuro, los resultados son altamente satisfactorios. Se ha conseguido que el sistema sea capaz de generar automáticamente los distintos movimientos y posiciones de cámara sin necesidad de realizar manualmente diferentes configuraciones de cámara previas según el entorno, y se ha constatado el beneficio visual que aporta el uso de las técnicas cinematográficas, y principalmente el aspecto emotivo de éstas, a la hora de gestionar las cámaras.

Otro de los aspectos interesantes encontrados en el desarrollo del trabajo se basa en la utilización del guión como método de obtención de información para la gestión de la cámara, ya que aunque la información obtenida a partir de la estructura y el texto del guión es de gran utilidad para la gestión de las cámaras, una análisis más en profundidad del guión con las nuevas técnicas que están apareciendo en los últimos años en el campo del Procesamiento del Lenguaje Natural abriría un gran abanico de posibilidades que podrían proporcionar mucha más información, y mucho más relevante, para la gestión y el control de las cámaras en los entornos virtuales, así como para otros aspectos como la gestión de luces, la distribución y acciones de los personajes, etc.

La independencia del sistema del aspecto geométrico, aunque suponiendo una de las debilidades principales propuestas como trabajo futuro a solventar, supone una interesante característica del sistema, ya que permite ser incorporado a cualquier entorno virtual, previa adaptación de éste al algoritmo, y que abre la puerta a la evaluación del sistema con otro tipo de entornos utilizados en diversos campos como la narratología interactiva o el mundo de los videojuegos.

## 6.2. Trabajo futuro

El trabajo aquí presentado se pretende que sea el inicio de un proyecto más ambicioso en el que el sistema no sólo tenga el control de las cámaras, sino que además pueda gestionar el sistema de luces, o las acciones y movimientos de los propios personajes, pudiendo conseguir de esta manera una mayor calidad visual en los entornos virtuales actuales. Para poder lograr este propósito, lo primero que se debe mejorar son los distintos inconvenientes detectados a lo largo de este trabajo, así como evaluar el sistema con un mayor número ejemplos para poder determinar con amplio grado de certeza la eficiencia del sistema.

A lo largo de las siguientes secciones se van a proponer diferentes mejoras

que serán desarrolladas y evaluadas en un futuro próximo para intentar mejorar el sistema, supliendo las carencias detectadas, y pudiendo de esta manera llegar al objetivo final planteado.

### 6.2.1. Incorporación del aspecto geométrico

Como se ha adelantado en el capítulo anterior, uno de los principales problemas detectados es el desconocimiento, por parte del algoritmo, de la disposición geométrica de los actores y del entorno en el que se encuentran, produciendo en ciertos casos configuraciones de cámara no acordes a dicha disposición, o situando la cámara en posiciones en las que un objeto obstruye lo que se desea filmar. Para intentar solucionar este tipo de situaciones se proponen varias técnicas, las cuales serán desarrolladas y evaluadas para determinar la aproximación más adecuada, tanto en resultados visuales como en coste computacional.

Una de las posibles opciones consiste en incorporar el aspecto geométrico de la escena en el algoritmo, pudiendo así el propio módulo lógico determinar si la configuración seleccionada es la más adecuada a la disposición de los objetos, o por el contrario, se deben modificar los parámetros de la configuración. El principal problema que se puede apreciar en esta opción es la sobrecarga computacional que supone para el sistema, puesto que los motores gráficos ya están optimizados para este tipo de tareas y supondría una ejecución paralela de las operaciones que el propio motor proporciona, con el consiguiente aumento en el consumo de memoria.

Otra de las opciones viables sería la inclusión de un conjunto de eventos en el módulo gráfico que determinara este tipo de situaciones a la hora de realizar una configuración propuesta, exponiendo el problema al módulo lógico, el cual a partir de los eventos recogidos podría determinar una configuración de cámara más acorde a la disposición de los objetos. Esta aproximación no supone un aumento sustancial de la memoria necesaria, pero sí, por el contrario, un aumento en la comunicación entre ambos módulos, con el consecuente incremento de carga computacional.

En ambas propuestas sería interesante la utilización de algún tipo de plantillas de situaciones límite que indicaran la imposibilidad de representar lo que está sucediendo de una forma óptima desde el punto de vista de las técnicas cinematográficas. En estos casos, sería deseable que el *Director* pudiera tener cierto control sobre los personajes, pudiendo desplazarlos hasta situaciones más apropiadas a las configuraciones implementadas.

### 6.2.2. Análisis de los diversos métodos para el cálculo de la intensidad

Otro punto que quizá habría precisado de un análisis mayor es el estudio de diferentes métodos para el cálculo de la intensidad emocional de las

etiquetas emocionales proporcionadas por **Emotag**. Para ellos, se pretende realizar un análisis exhaustivo de las diferentes teorías psicológicas existentes sobre la intensidad emocional, así como la evaluación de los diferentes trabajos y recursos semánticos existentes, como por ejemplo **WordNet Affect**.

Una vez realizado esto, si el resultado no fuera satisfactorio para el propósito del proyecto, se estudiaría la forma de generar un *corpus* propio de guiones basado en el programa **Emotag**, que permitiera realizar una evaluación más en profundidad de los resultados obtenidos, e incluir un método para el cálculo de la intensidad emocional.

### 6.2.3. Estudio de la aplicación de grafos a los sistemas de emociones

Como se ha dicho, otra de las deficiencias encontradas en el sistema se deriva de la limitación del vocabulario del *corpus* utilizado por el sistema **Emotag**, junto con la carencia encontrada de que no evalúa las posibles relaciones existentes entre las palabras desde el punto de vista emocional.

Para solventar esta limitación, se propone el estudio de la aplicación de la teoría de grafos a este tipo de sistemas, así como el análisis y la evaluación de los diferentes trabajos existentes en este área que hayan aplicado este tipo de técnicas, examinando los resultados obtenidos e intentando, en caso de que la investigación sea satisfactoria, desarrollar un prototipo basado en el programa **Emotag** que permita incorporar un nivel más de evaluación en el que se tengan en cuenta las diferentes relaciones de las palabras respecto a la intensidad emocional.

De esta manera, el sistema analizaría los diferentes textos hasta obtener una evaluación de la emoción inmersa en el mismo de cada palabra, al igual que se realiza ahora, añadiéndole en este punto una ponderación respecto de la relación con el resto de palabras de la frase, obteniendo así un sistema capaz de tener en cuenta las distintas relaciones entre las palabras a la hora de etiquetar un texto, y extendiéndose con **WordNet**, al igual que se realiza ahora, en caso de no disponer el *corpus* de alguna de las palabras del análisis.

Además, se intentará aumentar el número de textos etiquetados por personas, así como su ámbito, para la generación del *corpus*, pudiendo así disponer de un vocabulario más amplio en la lista LEW del programa.

### 6.2.4. Generación de transiciones suaves

Ya se ha comentado que, en algunos casos el sistema plantea transiciones bruscas entre configuraciones de cámara al responder a los diferentes eventos producidos en el entorno virtual. Aunque en ciertas ocasiones estos cambios pueden resultar interesantes para impactar al espectador, por regla general producen desconcierto y desorientación.



Para evitar este tipo de transiciones se podría introducir un análisis a la hora de seleccionar una nueva configuración de cámara que comprobara si el cambio entre la configuración actual y la seleccionada produciría una transición suave o no, generando en caso negativo una configuración intermedia entre ambas para evitar el movimiento brusco de la cámara.

### 6.2.5. Aumento del tipo de secuencias

Otro aspecto que podría enriquecer notablemente al sistema sería la posibilidad de poder identificar, por parte del *Analizador de Guiones*, y mediante técnicas de Procesamiento del Lenguaje Natural, otro tipo de secuencias más especializadas; como por ejemplo, secuencias románticas, de terror, de persecución, etc.

De este modo se plantea el estudio de diversos métodos de EI que permitan identificar patrones dentro de los textos de los guiones, a partir de un *corpus* previamente anotado por humanos con las diferentes secuencias, utilizando distintas técnicas de Aprendizaje Automático implementadas en numerosos sistemas de EI, pudiendo así realizar un clasificador de secuencias de guiones más específicas.

Aunque es cierto que los sistemas de clasificación actuales no suelen obtener resultados con un alto porcentaje de acierto, actualmente están surgiendo técnicas y recursos semánticos, como **WordNet** o **Google**, que podrían ser de gran interés para la realización de esta tarea, y que están empezando a utilizarse en diversos sistemas de EI.

### 6.2.6. Adaptación del algoritmo para entornos interactivos

Otro de los aspectos interesantes que se propone evaluar en un trabajo futuro es la posibilidad de adaptar el algoritmo generado a entornos interactivos, donde la gestión de cámara no se relega sólo a lo que está sucediendo en la historia, sino que se debe tener en cuenta la entrada al sistema proporcionada por el usuario en el control del avatar o personaje.

Una aproximación sencilla sería interpretar la escena tal y como se realiza actualmente, pero permitiendo que el usuario pudiera modificar las posiciones y los movimientos realizados por el algoritmo, de tal manera que el gestor de cámaras ejecutara las distintas configuraciones siempre y cuando el usuario no produjera ningún evento en la entrada. De esta forma el sistema le mostraría las conversaciones, o movimientos de los personajes, o las entradas y salidas de éstos en la escena, según las diferentes técnicas cinematográficas, relegando siempre el control al usuario cuando éste deseara visionar las escenas desde su perspectiva por medio de los controles de entrada.

Otra opción un poco más compleja sería la introducción de esa entrada en el algoritmo de gestión de cámaras, interpretándola y llegando a un compromiso entre la posición deseada por el usuario y la mejor configuración de

cámara seleccionada por el sistema. Esta aproximación debe ser evaluada cuidadosamente, ya que puede producir en el usuario la sensación de que el sistema no responde como él desea, pudiendo incluso llegar a disgustarle.

#### **6.2.7. Mejora del sistema de carga del módulo gráfico**

Como ya se ha adelantado, una característica interesante, aunque no aporta una mejora a los resultados finales, sería la realización de un sistema automático de carga de los objetos y acciones del módulo gráfico, el cual proporcionaría una interfaz más amigable y sencilla a la hora de generar la historia en el mundo virtual.

Una forma simple de realizar esta tarea sería mediante una interfaz que generará archivos de configuración en formato XML que entendiera el módulo gráfico, en los que se incluyeran todos los parámetros necesarios para su correcta carga en el sistema.

## Bibliografía

- [Amerson y Kime2001] Amerson, Daniel y Shaun Kime. 2001. Real-time cinematic camera control for interactive narratives. En *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, páginas 1–4, Stanford, CA, USA, March 26-28. AAAI Press.
- [Anscombe y Geach1970] Anscombe, E. y P. Geach. 1970. *Descartes Philosophical Writings*. Nelson: The Open University.
- [Arijon1991 (originally published 1976)] Arijon, Daniel. 1991 (originally published 1976). *Grammar of the Film Language*. Silman-James Press, September.
- [Aylett y Cavazza2001] Aylett, R. y M. Cavazza. 2001. Intelligent virtual environments - A state-of-the-art report. En D. Duke y R. Scopigno, editores, *STAR Proceedings of Eurographics 2001*, Manchester, UK, Septiembre. Eurographics Association.
- [Bares et al.2000] Bares, William, Scott McDermott, Christina Boudreaux, y Somying Thainimit. 2000. Virtual 3D Camera Composition from Frame Constraints. En *Proceedings of 8th ACM International Conference on Multimedia*, páginas 177–186, Los Angeles, CA, USA, October 30 - November 3.
- [Bares, Grégoire, y Lester1998] Bares, William H., Joël P. Grégoire, y James C. Lester. 1998. Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds. páginas 1101–1106, July 26-30.
- [Bares, Thainimit, y McDermott2000] Bares, William H., Somying Thainimit, y Scott McDermott. 2000. A model for constraint-based camera planning. En *Proceedings of AAAI Spring Symposium on Smart Graphics*, páginas 84–91, Stanford, CA, USA, March 20-22.
- [Benhamou et al.2004] Benhamou, Frédéric, Frédéric Goualard, Éric Languéno, y Marc Christie. 2004. Interval constraint solving for camera control and motion planning. *ACM Trans. Comput. Logic*, 5(4):732–767.
- [BLINN1988] BLINN, J. 1988. Where am i? what am i looking at? *IEEE Comput. Graph. Appl.*, 8(4):76–81.
- [Bourne y Sattar2005] Bourne, Owen y Abdul Sattar. 2005. Applying Constraint Weighting to Autonomous Camera Control. En *Proceedings of The First Artificial Intelligence and Interactive Digital Entertainment Conference*, páginas 3–8, Marina Del Ray, CA, USA, June 1-3. AAAI Press.

- [Bourne, Sattar, y Goodwin2008] Bourne, Owen, Abdul Sattar, y Scott Goodwin. 2008. A constraint-based autonomous 3d camera system. *Constraints*, 13(1-2):180–205.
- [Brooks1988] Brooks, F. P. 1988. Grasping reality through illusion—interactive graphics serving science. En *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, páginas 1–11, New York, NY, USA. ACM.
- [Burtnyk et al.2002] Burtnyk, N., A. Khan, G. Fitzmaurice, R. Balakrishnan, y G. Kurtenbach. 2002. Stylecam: Interactive stylized 3d navigation using integrated spatial and temporal controls.
- [Chen, Mountford, y Sellen1988] Chen, Michael, S. Joy Mountford, y Abigail Sellen. 1988. A study in interactive 3-d rotation using 2-d control devices. En *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, páginas 121–129, New York, NY, USA. ACM.
- [Christianson et al.1996] Christianson, David B., Sean E. Anderson, Li wei He, David Salesin, Daniel S. Weld, y Michael F. Cohen. 1996. Declarative camera control for automatic cinematography. páginas 148–155.
- [Christie y Languénou2003] Christie, Marc y Éric Languénou. 2003. A Constraint-based Approach to Camera Path Planning. En Andreas Butz Antonio Krüger, y Patrick Olivier, editores, *Proceedings of 3rd International Symposium on Smart Graphics*, páginas 172–181, Heidelberg, Germany, July 2-4. Springer.
- [Christie, Languénou, y Granvilliers2002] Christie, Marc, Eric Languénou, y Laurent Granvilliers. 2002. Modeling camera control with constrained hypertubes. En *CP '02: Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*, páginas 618–632, London, UK. Springer-Verlag.
- [Christie et al.2005] Christie, Marc, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, y Jonathan Pickering. 2005. Virtual camera planning: A survey. En *Proceedings of 5th International Symposium on SmartGraphics*, Frauenwoerth, Germany, August 22-24. Springer.
- [Christie y Normand2005] Christie, Marc y Jean-Marie Normand. 2005. A semantic space partitioning approach to virtual camera composition. *Computer Graphics Forum*, 24(3):247–256.
- [Christie y Olivier2006] Christie, Marc y Patrick Olivier. 2006. Camera control in computer graphics. páginas 89–113.

- [Cohen y Sarawagi2004] Cohen, W. y S. Sarawagi. 2004. Exploiting dictionaries in named entity extraction: Combining semi-markov extraction processes and data integration methods.
- [Courty y Marchand2001] Courty, N. y E. Marchand. 2001. Computer animation: a new application for image-based visual servoing.
- [Cowie y Cornelius2003] Cowie, Roddy y Randolph R. Cornelius. 2003. Describing the emotional states that are expressed in speech. *Speech Commun.*, 40(1-2):5–32.
- [Cunningham1999] Cunningham, H. 1999. Jape – a java annotation patterns engine.
- [Dillenbourg, Schneider, y Synteta2007] Dillenbourg, Pierre, Daniel Schneider, y Paraskevi Synteta. 2007. Virtual learning environments, Noviembre 23.
- [Dingli, Ciravegna, y Wilks2004] Dingli, Alexiei, Fabio Ciravegna, y Yorick Wilks. 2004. Automatic semantic annotation using unsupervised, Abril 07.
- [Drucker, Galyean, y Zeltzer1992] Drucker, Steven M., Tinsley A. Galyean, y David Zeltzer. 1992. Cinema: A system for procedural camera movements. En *Proceedings of 1992 Symposium on Interactive 3D Graphics*, páginas 67–70, Cambridge, MA, USA, March 29 - April 1.
- [Drucker y Zeltzer1994] Drucker, Steven M. y David Zeltzer. 1994. Intelligent Camera Control in a Virtual Environment. En *Proceedings of Graphics Interface '94*, páginas 190–199, Banff, Alberta, Canada, May 18-20.
- [Drucker y Zeltzer1995] Drucker, Steven M. y David Zeltzer. 1995. CamDroid: A System for Implementing Intelligent Camera Control. En *Proceedings of 1995 Symposium on Interactive 3D Graphics*, páginas 139–144, Monterey, CA, USA, April 9-12.
- [Ekman1992] Ekman, Paul. 1992. Are there basic emotions? *Psychological Review*, 99(3):550–553, Julio.
- [Evans, Tanner, y Wein1981] Evans, Kenneth B., Peter P. Tanner, y Marcell Wein. 1981. Tablet-based valuator that provide one, two, or three degrees of freedom. En *SIGGRAPH '81: Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, páginas 91–97, New York, NY, USA. ACM.
- [Fisher et al.1986] Fisher, S. S., M. McGreevy, J. Humphries, y W. Robinett. 1986. Virtual environment display system. En *SI3D '86: Proceedings of*

- the 1986 workshop on Interactive 3D graphics*, páginas 77–87, New York, NY, USA. ACM.
- [Francisco y Hervás2007] Francisco, V. y R. Hervás. 2007. EmoTag: Automated Mark Up of Affective Information in Texts. En Corina Forascu Oana Postolache Georgiana Puscasu, y Cristina Vertan, editores, *Proceedings of the Doctoral Consortium in EUROLAN 2007 Summer School*, páginas 5–12, Iasi, Romania, July–August.
- [Frederick P. Brooks1987] Frederick P. Brooks, Jr. 1987. Walkthrough—a dynamic graphics system for simulating virtual buildings. En *SI3D '86: Proceedings of the 1986 workshop on Interactive 3D graphics*, páginas 9–21, New York, NY, USA. ACM.
- [Halper, Helbing, y Strothotte2001] Halper, Nicolas, Ralf Helbing, y Thomas Strothotte. 2001. A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence. En A. Chalmers y T.-M. Rhyne, editores, *EG 2001 Proceedings*, volumen 20(3). Blackwell Publishing, páginas 174–183.
- [Halper y Olivier2000] Halper, Nicolas y Patrick Olivier. 2000. CAMPLAN: A Camera Planning Agent. En *Proceedings of AAAI Spring Symposium on Smart Graphics*, páginas 92–100, Stanford, CA, USA, March 20–22.
- [Handsuh, Staab, y Ciravegna2002] Handsuh, Siegfried, Steffen Staab, y Fabio Ciravegna. 2002. S-CREAM – semi-automatic CREation of metadata. *Lecture Notes in Computer Science*, 2473:358–??
- [Hanson1992] Hanson, Andrew J. 1992. The rolling ball. páginas 51–60, San Diego, CA, USA. Academic Press Professional, Inc.
- [Hanson y Wernert1997] Hanson, Andrew J. y Eric A. Wernert. 1997. Constrained 3D navigation with 2D controllers. En *IEEE Visualization*, páginas 175–182.
- [Hawkins2005] Hawkins, Brian. 2005. *Real-Time Cinematography for Games*. Charles River Media, January 28.
- [Hornung, Lakemeyer, y Trogemann2003] Hornung, Alexander, Gerhard Lakemeyer, y Georg Trogemann. 2003. An Autonomous Real-Time Camera Agent for Interactive Narratives and Games. En *Proceedings of Fourth International Workshop on Intelligent Agents (IVA 2003)*, páginas 236–243, Kloster Irsee, Germany, September 15–17. Springer.
- [Hutchinson, Hager, y Corke1996] Hutchinson, S. A., G. D. Hager, y P. I. Corke. 1996. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, Octubre.

- [Ibáñez-Martínez, Delgado-Mata, y Aylett2006] Ibáñez-Martínez, Jesús, Carlos Delgado-Mata, y Ruth Aylett. 2006. Virtual environments: A multi-disciplinary field. *UPGRADE: The European Journal for the Informatics Professional*, Vol. VII.
- [Izard1977] Izard, C.E. 1977. Human emotions. *Plenum Press, New York*.
- [James1884] James, William. 1884. *What is emotion?*, volumen Mind 9.
- [Jardillier y Languénou1998] Jardillier, Frank y Eric Languénou. 1998. Screen-space constraints for camera movements: the virtual cameraman. *Comput. Graph. Forum*, 17(3):175–186.
- [Jung, Paik, y Kim1998] Jung, Moon-Ryul, Doowon Paik, y Donghyun Kim. 1998. A camera control interface based on the visualization of subspaces of the 6d motion space of the camera. En *PG '98: Proceedings of the 6th Pacific Conference on Computer Graphics and Applications*, página 198, Washington, DC, USA. IEEE Computer Society.
- [Kahkonen y Whyte2003] Kahkonen, Kalle y Jennifer Whyte. 2003. Industrial applications of virtual reality in architecture and construction.
- [Khan et al.2005] Khan, Azam, Ben Komalo, Jos Stam, George Fitzmaurice, y Gordon Kurtenbach. 2005. Hovercam: interactive 3d navigation for proximal object inspection. En *I3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, páginas 73–80, New York, NY, USA. ACM.
- [Kiryakov et al.2004] Kiryakov, Atanas, Borislav Popov, Ivan Terziev, Dimitar Manov, y Damyan Ognyanoff. 2004. Semantic annotation, indexing, and retrieval. *J. Web Sem*, 2(1):49–79.
- [Kneafsey y McCabe2005] Kneafsey, James y Hugh McCabe. 2005. Camerabots: Cinematography for games with non-player characters as camera operators. En *DIGRA Conf*.
- [Kogut2001] Kogut, Paul. 2001. Aerodaml: Applying information extraction to generate daml annotations from web pages. En *First International Conference on Knowledge Capture (K-CAP 2001). Workshop on Knowledge Markup and Semantic Annotation*.
- [Lang1980] Lang, P.J. 1980. Behavioral treatment and bio-behavioral assessment: Computer applications. *Ablex Publishing, Norwood*.
- [Lang y Bradley1999] Lang, P.J. y M.M. Bradley. 1999. Affective norms for english words (anew). En *Technical Manual and Affective Ratings*.

- [Li y Hsu2004] Li, Tsai-Yen y Shu-Wei Hsu. 2004. An intelligent 3d user interface adapting to user control behaviors. En *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, páginas 184–190, New York, NY, USA. ACM.
- [Lin, Shih, y Tsai2004] Lin, Ting-Chieh, Zen-Chung Shih, y Yu-Ting Tsai. 2004. Cinematic camera control in 3d computer games. En *Proceedings of 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2004)*, páginas 289–296, University of West Bohemia, Campus Bory, Plzen-Bory, Czech Republic, February 2-6.
- [Mackinlay, Robertson, y Card1991] Mackinlay, Jock D., George G. Robertson, y Stuart K. Card. 1991. Rapid controlled movement through virtual 3d workspaces (videotape). En *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, páginas 455–456, New York, NY, USA. ACM.
- [Marchand y Hager] Marchand, E. y G. Hager. Dynamic sensor planning in visual servoing. páginas 1988–1993.
- [Marchand y Courty2002] Marchand, Éric y Nicolas Courty. 2002. Controlling a camera in a virtual environment. *The Visual Computer*, 18(1):1–19.
- [Mascelli1965] Mascelli, J.V. 1965. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Hollywood.
- [Michelson y Knoblock2005] Michelson, Matthew y Craig A. Knoblock. 2005. Semantic annotation of unstructured and ungrammatical text. En Leslie Pack Kaelbling y Alessandro Saffioti, editores, *IJCAI*, páginas 1091–1098. Professional Book Center.
- [Millerson1983] Millerson, Gerrald. 1983. *Técnicas de Realización y Producción en Televisión*.
- [Nielson y Dan R. Olsen1987] Nielson, Gregory M. y Jr. Dan R. Olsen. 1987. Direct manipulation techniques for 3d objects using 2d locator devices. En *SI3D '86: Proceedings of the 1986 workshop on Interactive 3D graphics*, páginas 175–182, New York, NY, USA. ACM.
- [Ortony y Turner1990] Ortony, A. y T. J. Turner. 1990. What's basic about basic emotions? *Psychol Rev*, 97(3):315–331, July.
- [Ortony, Clore, y Collins1988] Ortony, Andrew, Gerald Clore, y Allan Collins. 1988. *The Cognitive Structure of Emotions*. Cambridge University Press.



- [Parrott2001] Parrott, W.G. 2001. *Emotions in Social Psychology: Essential Readings*. Psychology Press, Philadelphia.
- [Peinado2006] Peinado, Federico. 2006. Interactive digital storytelling: Automatic direction of virtual environments. *Upgrade. Monograph: Virtual Environments*, VII(2):42–46.
- [Phillips, Badler, y Granieri1992] Phillips, Cary B., Norman I. Badler, y John Granieri. 1992. Automatic viewing control for 3d direct manipulation. En *Proceedings of 1992 Symposium on Interactive 3D Graphics*, páginas 71–74, Cambridge, MA, USA, March 29 - April 1.
- [Pickering2002] Pickering, Jonathan H. 2002. Intelligent camera planning for computer graphics.
- [Plutchik1980] Plutchik, R. 1980. *A general Psycho Evolutionary Theory of Emotion*. Emotion: Theory, Research, and Experience.
- [Popov et al.2004] Popov, Borislav, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, y Angel Kirilov. 2004. Kim – a semantic platform for information extraction and retrieval. *Nat. Lang. Eng.*, 10(3-4):375–392.
- [Reeve y Han2005] Reeve, Lawrence y Hyoil Han. 2005. Survey of semantic annotation platforms. En *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, páginas 1634–1638, New York, NY, USA. ACM.
- [Robinson y Mania2007] Robinson, Andrew y Katerina Mania. 2007. Technological research challenges of flight simulation and flight instructor assessments of perceived fidelity. *Simul. Gaming*, 38(1):112–135.
- [Russell1980] Russell, J. A. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.
- [Sbretchs et al.2002] Sbretchs, M., D. Clawson, G. Higgins, C.E. Lathan, y M. Tracey. 2002. Using virtual environments as training simulators: Measuring transfer. In *Handbook of Virtual Environments K Stanney, Ed., Lawrence Erlbaum Associates*, páginas 403–415.
- [Shoemake1992] Shoemake, K. 1992. Arcball: A user interface for specifying three-dimensional orientation using a mouse. En *Proc. of the Graphics Interface '92*, páginas 151–156, Vancouver, Canada.
- [Shoemake1994] Shoemake, Ken. 1994. Arcball rotation control. páginas 175–192.
- [Sutherland1963] Sutherland, I. 1963. *A Man-Machine Graphical Communication System*. Ph.D. tesis, MIT, Enero.

- [Tan, Robertson, y Czerwinski2000] Tan, Desney S, George G Robertson, y Mary Czerwinski. 2000. Exploring 3d navigation: Combining speed-coupled flying with orbiting. En *CHI 2001 Conference on Human Factors in Computing Systems*, Seattle, WA.
- [Tomlinson, Blumberg, y Nain2000] Tomlinson, Bill, Bruce Blumberg, y Delphine Nain. 2000. Expressive Autonomous Cinematography for Interactive Virtual Environments. En *Proceedings of Fourth International Conference on Intelligent Agents*, páginas 317–324, Barcelona, Spain, June 3-7.
- [Torres-Guizar2006] Torres-Guizar, Daniel. 2006. A quick look at the video-game industry. *Upgrade. Monograph: Virtual Environments*, VII(2):33–37.
- [Turner et al.1991] Turner, Russell, Francis Balaguer, Enrico Gobetti, y Daniel Thalmann. 1991. Physically-based interactive camera motion control using 3D input devices. En N. M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena (Proceedings of CG International '91)*, páginas 135–145. Springer-Verlag.
- [Ware y Osborne1990] Ware, Colin y Steven Osborne. 1990. Exploration and virtual camera control in virtual three dimensional environments. En *Proceedings of 1990 Symposium on Interactive 3D Graphics*, páginas 175–183, Snowbird, UT, USA, March. ACM Press.
- [Watson y Tellegen1985] Watson, D. y A. Tellegen. 1985. Toward a consensual structure of mood. *Psychological bulletin*, 98(2):219–235, September.
- [wei He, Cohen, y Salesin1996] wei He, Li, Michael F. Cohen, y David H. Salesin. 1996. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. En *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, páginas 217–224, New York, NY, USA. ACM.
- [Wood et al.1997] Wood, DanielÑ., Adam Finkelstein, John F. Hughes, Craig E. Thayer, y David H. Salesin. 1997. Multiperspective panoramas for cel animation. *Computer Graphics*, 31(Annual Conference Series):243–250.
- [Xiao y Hubbard1998] Xiao, Dongbo y Roger Hubbard. 1998. Navigation guided by artificial force fields. En *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, páginas 179–186, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Zancanaro, Rocchi, y Stock2003] Zancanaro, Rocchi, y Stock. 2003. Automatic video composition.

- [Zelevnik y Forsberg1999] Zelevnik, Robert C. y Andrew S. Forsberg. 1999. Unicam - 2d gestural camera controls for 3d environments. En *Symposium on Interactive 3D Graphics*, páginas 169–173.



---

## Apéndice A

# Ejemplos de reglas en el sistema

A lo largo de este apéndice se van a exponer algunos ejemplos de las diferentes reglas implementadas en el sistema.

```
rule "Action_Motion_Static_Init"
  salience 75
  when
    a:ActionSeq( (motion==true));
    not CameraConfiguration();
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setTime(3000);
    cameraConfiguration.setActualRule("Action_Motion_Static_Init");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Motion_Static_Init");
  end
```

Figura A.1: Ejemplo 1 de las reglas de las secuencias de acción del sistema.

```

rule "Action_Motion_Movement_LONG_SHOT"
salience 75
when
    a:ActionSeq((motion==true) && ((emotionalIntensity(>=00 && <30)) ||
(emotionalIntensity(<=-00 && >-30))));
    CameraConfiguration(actualRule=="Action_Motion_Static_Init");
then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    Movement mov = new PanMov(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setMovement(mov);
    cameraConfiguration.setPrincipalObjects(a.getObjectsInMotion());
    cameraConfiguration.setLookAtMovement(0);
    cameraConfiguration.setTime(6000);
    cameraConfiguration.setActualRule("Action_Motion_Movement_LONG_SHOT");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Motion_Movement_LONG_SHOT");
end

rule "Action_Motion_Movement_LONG_SHOT2"
salience 75
when
    a:ActionSeq((motion==true) && ((emotionalIntensity(>=00 && <30)) ||
(emotionalIntensity(<00 && >-30))));
    c:CameraConfiguration((actualRule=="Action_Motion_FromStatic") ||
(actualRule=="Action_Motion_Movement_LONG_SHOT")) &&
(numPrincipalObjects>(lookAtMovement+1));
then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    Movement mov = new PanMov(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setMovement(mov);
    cameraConfiguration.setPrincipalObjects(a.getObjectsInMotion());
    cameraConfiguration.setLookAtMovement(c.getLookAtMovement()+1);
    cameraConfiguration.setTime(6000);
    cameraConfiguration.setActualRule("Action_Motion_Movement_LONG_SHOT2");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Motion_Movement_LONG_SHOT2");
end

```

Figura A.2: Ejemplo 2 de las reglas de las secuencias de acción del sistema.

```

rule "Action_Motion_Movement_FULL_SHOT2"
salience 75
when
    a:ActionSeq((motion==true) && ((emotionalIntensity(>=30 && <60)) ||
(emotionalIntensity(<=-30 && >-60))));
    c:CameraConfiguration((actualRule=="Action_Motion_FromStatic") ||
(actualRule=="Action_Motion_Movement_FULL_SHOT")) &&
(numPrincipalObjects>(lookAtMovement+1));
then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    Movement mov = new PanMov(new Distance(Shots.FULL_SHOT));
    cameraConfiguration.setMovement(mov);
    cameraConfiguration.setPrincipalObjects(a.getObjectsInMotion());
    cameraConfiguration.setLookAtMovement(c.getLookAtMovement()+1);
    cameraConfiguration.setTime(6000);
    cameraConfiguration.setActualRule("Action_Motion_Movement_FULL_SHOT2");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Motion_Movement_FULL_SHOT2");
end

rule "Action_Motion_Movement_MEDIUM_FULL_SHOT"
salience 75
when
    a:ActionSeq((motion==true) && ((emotionalIntensity(>=60 && <100)) ||
(emotionalIntensity(<=-60 && >-100))));
    CameraConfiguration(actualRule=="Action_Motion_Static_Init");
then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    Movement mov = new PanMov(new Distance(Shots.MEDIUM_FULL_SHOT));
    cameraConfiguration.setMovement(mov);
    cameraConfiguration.setPrincipalObjects(a.getObjectsInMotion());
    cameraConfiguration.setLookAtMovement(0);
    cameraConfiguration.setTime(6000);
    cameraConfiguration.setActualRule("Action_Motion_Movement_MEDIUM_FULL_SHOT");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Motion_Movement_MEDIUM_FULL_SHOT");
end

```

Figura A.3: Ejemplo 3 de las reglas de las secuencias de acción del sistema.

```

rule "Action_Static_FromMotion1And2P"
  salience 75
  when
    a:ActionSeq((numActors(==1 || ==2)) && (motion==false));
    CameraConfiguration(actualRule matches "Action_Motion\S*");
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setTime(5000);
    cameraConfiguration.setAngle(10);
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setActualRule("Action_Static_FromMotion1And2P");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Static_FromMotion1And2P");
  end

rule "Action_Static_Init1And2P"
  salience 75
  when
    a:ActionSeq((numActors(==1 || ==2)) && (motion==false));
    not CameraConfiguration();
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setTime(5000);
    cameraConfiguration.setAngle(10);
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setActualRule("Action_Static_Init1And2P");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Static_Init1And2P");
  end

```

Figura A.4: Ejemplo 4 de las reglas de las secuencias de acción del sistema.



```

rule "Action_Static_2+MEDIUM_SHOT2"
salience 50
when
    a:ActionSeq{(numActors(>=2)) && ((emotionalIntensity(>=60 && <80))
    || (emotionalIntensity(<=-60 && >-80)))};
    c:CameraConfiguration{(actualRule=="Action_Static_MEDIUM_SHOT" ||
    actualRule=="Action_Static_2+MEDIUM_SHOT2") && (numPrincipalObjects>(lookAtShot+1))};
then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.MEDIUM_SHOT));
    cameraConfiguration.setTime(6000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setLookAtShot(c.getLookAtShot()+1);
    cameraConfiguration.setActualRule("Action_Static_2+MEDIUM_SHOT2");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Static_2+MEDIUM_SHOT2");
end

rule "Action_Static_AMERICAN_SHOT"
salience 25
when
    a:ActionSeq{(numActors(>=1)) && ((emotionalIntensity(>=40 && <60))
    || (emotionalIntensity(<=-40 && >-60)))};
    CameraConfiguration(actualRule matches "Action_Static\S*");
then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.AMERICAN_SHOT));
    cameraConfiguration.setTime(5000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setLookAtShot(0);
    cameraConfiguration.setActualRule("Action_Static_AMERICAN_SHOT");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Static_AMERICAN_SHOT");
end

```

Figura A.5: Ejemplo 5 de las reglas de las secuencias de acción del sistema.

```

rule "Action_Static_LONG_SHOT_GROUP"
  salience 25
  when
    a:ActionSeq((numActors(>=1)) && ((emotionalIntensity(>=00 && <30)) ||
      (emotionalIntensity(<=-00 && >-30))));
    CameraConfiguration{(actualRule=="Action_Static_2+MEDIUM_CLOSE_SHOT2")
    || (actualRule=="Action_Static_2+MEDIUM_SHOT2") ||
    (actualRule=="Action_Static_2+AMERICAN_SHOT2")
    || (actualRule=="Action_Static_2+MEDIUM_FULL_SHOT2" ||
    (actualRule=="Action_Static_2FULL_SHOT2"))};
    CameraConfiguration(numPrincipalObjects==(lookAtShot+1));
    CameraConfiguration(actualRule matches "Action_Static\S*");
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setCameraPosition(new Position(CameraPosition.APEX));
    cameraConfiguration.setTime(8000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setActualRule("Action_Static_LONG_SHOT_GROUP");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Static_LONG_SHOT_GROUP");
  end

rule "Action_Static_FULL_SHOT_GROUP"
  salience 25
  when
    a:ActionSeq((numActors(>=1)) && ((emotionalIntensity(>=30 && <60))
    || (emotionalIntensity(<=-30 && >-60))));
    CameraConfiguration{(actualRule=="Action_Static_MEDIUM_SHOT2") ||
    (actualRule=="Action_Static_2+AMERICAN_SHOT2") ||
    (actualRule=="Action_Static_2+MEDIUM_FULL_SHOT2")
    || (actualRule=="Action_Static_2+FULL_SHOT2" ||
    (actualRule=="Action_Static_2LONG_SHOT2"))};
    CameraConfiguration(numPrincipalObjects==(lookAtShot+1));
    CameraConfiguration(actualRule matches "Action_Static\S*");
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.FULL_SHOT));
    cameraConfiguration.setCameraPosition(new Position(CameraPosition.APEX));
    cameraConfiguration.setTime(8000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(a.getActors());
    cameraConfiguration.setActualRule("Action_Static_FULL_SHOT_GROUP");
    listCamConf.add(cameraConfiguration);
    System.out.println("Action_Static_FULL_SHOT_GROUP");
  end

```

Figura A.6: Ejemplo 6 de las reglas de las secuencias de acción del sistema.

```

rule "Dialog_Static_Init2P_LONG_SHOT"
  salience 50
  when
    d:DialogSeq((numActors(==2)) && (motion==false)&&
      ((emotionalIntensity(>=00 && <30)) || (emotionalIntensity(<=-00 && >-30))));
    not CameraConfiguration();
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.LONG_SHOT));
    cameraConfiguration.setCameraPosition(new Position(CameraPosition.APEX));
    cameraConfiguration.setTime(4000);
    cameraConfiguration.setAngle(10);
    cameraConfiguration.setPrincipalObjects(d.getActors());
    cameraConfiguration.setActualRule("Dialog_Static_Init2P_LONG_SHOT");
    listCamConf.add(cameraConfiguration);
    System.out.println("Dialog_Static_Init2P_LONG_SHOT");
  end

rule "Dialog_Static_Init2P_FULL_SHOT"
  salience 50
  when
    d:DialogSeq((numActors(==2)) && (motion==false)&&
      ((emotionalIntensity(>=30 && <60)) || (emotionalIntensity(<=-30 && >-60))));
    not CameraConfiguration();
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.FULL_SHOT));
    cameraConfiguration.setCameraPosition(new Position(CameraPosition.APEX));
    cameraConfiguration.setTime(4000);
    cameraConfiguration.setAngle(10);
    cameraConfiguration.setPrincipalObjects(d.getActors());
    cameraConfiguration.setActualRule("Dialog_Static_Init2P_FULL_SHOT");
    listCamConf.add(cameraConfiguration);
    System.out.println("Dialog_Static_Init2P_FULL_SHOT");
  end

```

Figura A.7: Ejemplo 1 de las reglas de las secuencias de diálogo del sistema.

```

rule "Dialog_Static_Active_FULL_CLOSE_UP"
  salience 65
  when
    d:DialogSeq((numActors(==2)) && (motion==false) &&
      (dialogActive==true) && ((actorIntesitySpeech>=80) || (actorIntesitySpeech<=-80)));
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.FULL_CLOSE_UP));
    cameraConfiguration.setCameraPosition(new Position(CameraPosition.INTERNAL));
    cameraConfiguration.setTime(7000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(d.getActorSpeakinInArray());
    cameraConfiguration.setActualRule("Dialog_Static_Active_FULL_CLOSE_UP");
    listCamConf.add(cameraConfiguration);
    System.out.println("Dialog_Static_Active_FULL_CLOSE_UP");
  end

rule "Dialog_Static_Active_2+PFULL_CLOSE_UP2"
  salience 75
  when
    d:DialogSeq((numActors(==2)) && (motion==false) &&
      (dialogActive==true) && ((actorIntesitySpeech>=80) ||
      (actorIntesitySpeech<=-80)));
    c:CameraConfiguration((actualRule=="Dialog_Static_Active_FULL_CLOSE_UP")
      && (numPrincipalObjects>(lookAtShot)));
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.FULL_CLOSE_UP));
    cameraConfiguration.setCameraPosition(new Position(CameraPosition.INTERNAL));
    cameraConfiguration.setTime(4000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(d.getOthersActors());
    cameraConfiguration.setLookAtShot(0);
    cameraConfiguration.setActualRule("Dialog_Static_Active_2+PFULL_CLOSE_UP2");
    listCamConf.add(cameraConfiguration);
    System.out.println("Dialog_Static_Active_2+PFULL_CLOSE_UP2");
  end

```

Figura A.8: Ejemplo 2 de las reglas de las secuencias de diálogo del sistema.

```

rule "Dialog_Static_NoActive_FULL_CLOSE_UP"
    salience 75
    when
        d:DialogSeq((numActors==2)) && (motion==false) && (dialogActive==false)
        && ((emotionalIntensity>=80) || (emotionalIntensity<=-80));
    then
        CameraConfiguration cameraConfiguration = new CameraConfiguration();
        cameraConfiguration.setDistance(new Distance(Shots.FULL_CLOSE_UP));
        cameraConfiguration.setCameraPosition(new Position(CameraPosition.INTERNAL));
        cameraConfiguration.setTime(7000);
        cameraConfiguration.setAngle(0);
        cameraConfiguration.setPrincipalObjects(d.getActors());
        cameraConfiguration.setLookAtShot(0);
        cameraConfiguration.setActualRule("Dialog_Static_NoActive_FULL_CLOSE_UP");
        listCamConf.add(cameraConfiguration);
        System.out.println("Dialog_Static_NoActive_FULL_CLOSE_UP");
    end

rule "Dialog_Static_NoActive_2+PFULL_CLOSE_UP2"
    salience 65
    when
        d:DialogSeq((numActors==2)) && (motion==false) && (dialogActive==false) &&
        ((emotionalIntensity>=70) || (emotionalIntensity<=-70));
        c:CameraConfiguration((actualRule=="Dialog_Static_NoActive_FULL_CLOSE_UP"
        || actualRule=="Dialog_Static_NoActive_2+PFULL_CLOSE_UP2") &&
        (numPrincipalObjects>(lookAtShot+1)));
    then
        CameraConfiguration cameraConfiguration = new CameraConfiguration();
        cameraConfiguration.setDistance(new Distance(Shots.FULL_CLOSE_UP));
        cameraConfiguration.setCameraPosition(new Position(CameraPosition.INTERNAL));
        cameraConfiguration.setTime(7000);
        cameraConfiguration.setAngle(0);
        cameraConfiguration.setPrincipalObjects(d.getActors());
        cameraConfiguration.setLookAtShot(c.getLookAtShot()+1);
        cameraConfiguration.setActualRule("Dialog_Static_NoActive_2+PFULL_CLOSE_UP2");
        listCamConf.add(cameraConfiguration);
        System.out.println("Dialog_Static_NoActive_2+PFULL_CLOSE_UP2");
    end

```

Figura A.9: Ejemplo 3 de las reglas de las secuencias de diálogo del sistema.

```

rule "Dialog_Static_NoActive_MEDIUM_SHOT"
  salience 75
  when
    d:DialogSeq((numActors(==2)) && (motion==false)&& (dialogActive==false) &&
      ((emotionalIntensity(>=00 && <40)) || (emotionalIntensity(<=-00 && >-40))));
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.MEDIUM_SHOT));
    cameraConfiguration.setTime(5000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(d.getActors());
    cameraConfiguration.setLookAtShot(0);
    cameraConfiguration.setActualRule("Dialog_Static_NoActive_MEDIUM_SHOT");
    listCamConf.add(cameraConfiguration);
    System.out.println("Dialog_Static_NoActive_MEDIUM_SHOT");
  end

rule "Dialog_Static_NoActive_2+MEDIUM_SHOT2"
  salience 65
  when
    d:DialogSeq((numActors(==2)) && (motion==false)&& (dialogActive==false) &&
      ((emotionalIntensity(>=00 && <40)) || (emotionalIntensity(<=-00 && >-40))));
    c:CameraConfiguration((actualRule=="Dialog_Static_NoActive_MEDIUM_SHOT"
      || actualRule=="Dialog_Static_NoActive_2+MEDIUM_SHOT2") &&
      (numPrincipalObjects>(lookAtShot+1)));
  then
    CameraConfiguration cameraConfiguration = new CameraConfiguration();
    cameraConfiguration.setDistance(new Distance(Shots.MEDIUM_SHOT));
    cameraConfiguration.setTime(5000);
    cameraConfiguration.setAngle(0);
    cameraConfiguration.setPrincipalObjects(d.getActors());
    cameraConfiguration.setLookAtShot(c.getLookAtShot()+1);
    cameraConfiguration.setActualRule("Dialog_Static_NoActive_2+MEDIUM_SHOT2");
    listCamConf.add(cameraConfiguration);
    System.out.println("Dialog_Static_NoActive_2+MEDIUM_SHOT2");
  end

```

Figura A.10: Ejemplo 4 de las reglas de las secuencias de diálogo del sistema.

---

## Apéndice B

# Fragmento del guión de la película

INT. EDORAS, SLEEPING QUARTERS - NIGHT

Gimli is snoring in the room. The floor of the hall is scattered with sleeping bodies. Aragorn moves stealthily across the room and goes out through the door. Pippin turns over and wakes up. He tries to sleep obviously having an inward struggle with himself.

INT. EDORAS, GOLDEN HALL - NIGHT

Eowyn sleeps in the centre of the room next to the fire. Aragorn walks past her to the fire. He moves the logs around to encourage it to burn, he looks at her sleeping and walks over to her to pull her dress down to her ankle and her collar up to her neck

EOWYN

(with her eyes closed)  
What time is it?

ARAGORN

Not yet dawn.

Eowyn strokes the fur on her collar and withdraws his hand. Eowyn grabs his hand as he draws away. Aragorn turns back

and looks at her.

EOWYN

I dreamed I saw a great wave climbing over green  
lands and above the hills. I stood upon the brink.  
It was utterly dark in the abyss before my feet. A  
light shone behind me but I could not turn  
(he stoops down next to her and she looks at him  
upset) I could only stand there, waiting.  
(a tear runs down her face.)

ARAGORN

Night changes many thoughts. Sleep Eowyn.  
Sleep while you can.

Aragorn clutches her hands and then leaves her. Eowyn sleeps.  
Aragorn walks outside, starts to fill his pipe.



---

## Apéndice C

# Publicaciones

El desarrollo de este trabajo a dado lugar a las diferentes publicaciones relacionadas con el sistema que aquí se exponen.

- Jorge Carrillo de Albornoz, Luis Garmendia, Pablo Gervás, *Sistema automático de decisión del plano de cámara en videojuegos y entornos virtuales*. Segundo Congreso Internacional de Matemáticas en la Ingeniería y la Arquitectura, Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos, Universidad Politécnica de Madrid, 2008. Páginas: 219-230, ISBN: 978-84-7493-390-1.
- Carlos León, Jorge Carrillo de Albornoz, Pablo Gervás, *A Framework for Building Creative Objects From Heterogeneous Generation Systems*. 5th International Joint Workshop on Computational Creativity, Facultad de Informática de la UCM, Madrid, España. 17 - 19 de Septiembre del 2008.